Eletrônica Embarcada para Internet

KIT DE DESENVOLVIMENTO KPME-01

1. LISTA DE ABREVIAÇÕES.....5

2. LISTA DE ABREVIAÇÕES.....5

3. INTRODUÇÃO.....6

4. PLACA PME-10.....7

4.0 Conector para Placas de Expansão.....10

4.1 A PME-10 como Placa Filha.....11

4.2 A PME-10 com Placa Filha.....11

5. INTERFACES.....11

5.0 Ethernet.....11

5.1 RS232.....11

5.2 Conector ICSP.....12

5.3 Pinos Analógicos e Digitais de E/S.....12

6. ESPECIFICAÇÕES PME-10.....12

7. CONECTANDO A PLACA PME-10 PELA PRIMEIRA VEZ.....14

7.0 Precauções.....14

7.1 Conectando a PME-10 com um Host Local sem DHCP.....14

7.2 Alterando o endereço IP sem usar o DHCP.....18

7.3 Usando o DHCP.....28

8. DIMENSÕES DA PME-10.....29

9. ESQUEMAS.....30

10. CONFIGURAÇÃO DO PIC18F8720.....35

11. APÊNDICE A . CONFIGURANDO O WINDOWS XP COM IP FIXO.....41

12. APÊNDICE B.....42

13. APÊNDICE C. O SERVIDOR HTTP MICROCHIP.....44

14. APÊNDICE D. ENVIANDO E RECEBENDO MENSAGENS SMS ATRAVÉS DO CELULAR SIEMENS C55 (MODO PDU).....51

15. APÊNDICE E. DISPLAY DE LCD.....57

16. APÊNDICE F. CABO SERIAL CELULAR.....58

17. APÊNDICE G. CABO SERIAL PROGRAMAÇÃO.....58

18. APÊNDICE H. DIFERENÇAS PARA PROGRAMAR O PIC18F8720 E O PIC18F8621.....59

19. APÊNDICE I. TECLADO 60

20. APÊNDICE J. CABO DE REDE.....61

21. APÊNDICE K. ALTERAÇÕES NA VERSÃO DESTE MANUAL.....61

1. Lista de Abreviações

ARP – Address Resolution Protocol

CGI - Common Gateway Interface

DHCP – Dynamic Host Configuration Protocol

LAN – Local Area Network

LED – Light Emitting Diode

FTP – File Transfer Protocol

HTTP - Hypertext Transfer Protocol

ICMP – Internet Control Message Protocol

ICSP - In Circuit Serial Programmer

MAC – Medium Access Control

MPFS – Microchip File System

PDU – Packet Data Unit

PME-10 – Placa Microcontrolada com Ethernet 10 Mbps

RTOS – Real Time Operating System

SMS – Short Message Service

TCP/IP - Transmission Control Protocol/Internet Protocol

UCP – Unidade Central de Processamento

UDP- User Data-Gram Protocol

USART – Universal Serial Asynchronous Receiver Transmitter

2. Introdução

Esta documentação refere-se ao kit com conectividade *Ethernet/Internet* KPME-10 versão 1.0 (Figura 1) .

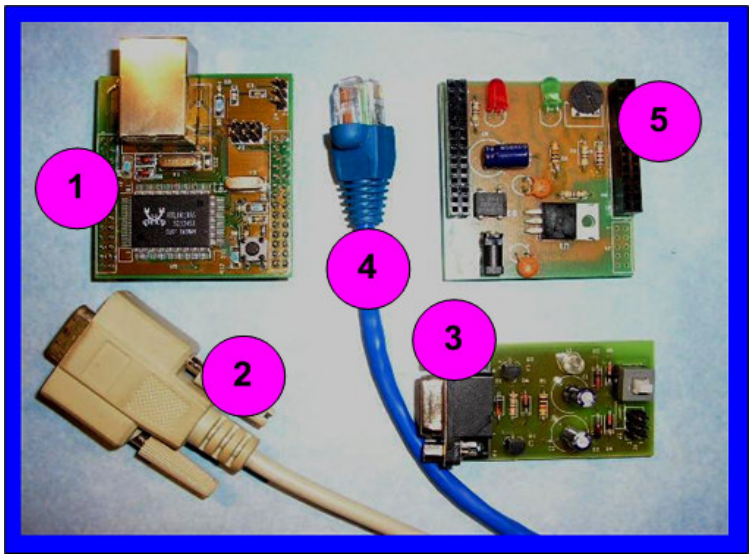


Figura 1 Itens que compõeo kit KPME-10

O kit contém os seguintes itens:

1. Placa PME-10 com conexão *Ethernet/Internet*.
2. Um cabo serial para a interface de programação da PME-10.
3. Uma interface de programação serial RS-232 para ICSP (*In Circuit Serial Programmer*) para gravação da memória Flash do microcontrolador contido na PME-10. Somente utilize uma interface serial RS232 verdadeira em seu PC. Conversores USB para serial provavelmente não funcionarão. Só programe a PME-10 com ela alimentada e não inverta o cabo que vai da interface de programação para a PME-10. Maus procedimentos causarão danos ao microcontrolador.

4. Um cabo de rede *crossover* (10 Base T) para conectar a placa PME-10 com um computador tipo PC.
5. Uma placa filha contendo fonte de +5VDC, LED, chave *push-button* e potenciômetro para demonstração de monitoração remota.
6. Um CD-ROM contendo: a pilha TCP/IP Microchip modificada, programa IC-PROG para gravação da memória Flash do microcontrolador PIC18F8720 presente na placa PME-10 e guia do usuário em formato PDF.
7. Guia do usuário.

O compilador C MCC18 (versão estudante) e o ambiente de desenvolvimento MPLAB IDE são propriedades da Microchip®. Eles deverão ser baixados do site www.microchip.com . Os links estão disponíveis no site www.2ei.com.br.

Leia a AN005 que se encontra no CD, Como Usar o ICPROG no Windows XP/NT/2000, para obter um tempo de gravação em torno de 35 segundos do microcontrolador da PME-10.

3. Placa PME-10

A PME-10 (Figura 2) é uma placa microcontrolada (UCP PIC 18F8720) com interfaces *Ethernet* 10 Mbps e serial RS-232. Ela pode ser adicionada a qualquer rede *Ethernet* 10/100 Mbps. Ela é fornecida com a pilha TCP/IP da Microchip modificada escrita na linguagem C, que pode ser compilada tanto pelo compilador Microchip PIC18 C como pelo compilador Hi Tech. Aplicações típicas incluem servidor *Web* HTTP, conversor de interface RS 232 para Ethernet, conversor de interface RS485 para *Ethernet*, controle remoto via *Web*, etc. Você pode armazenar suas páginas Web na memória FLASH do microcontrolador PIC18F8720 usando o cabo serial e interface de programação que acompanha o kit de desenvolvimento. Além da pilha TCP/IP da Microchip, a 2EI fornece funções de enviar e receber mensagens SMS via celular através da interface serial.

Para ver as páginas Web contidas na UCP PIC 18F8720 num navegador *Internet* basta conectar o cabo *crossover* fornecido neste *kit* ao seu computador pessoal com a devida configuração.

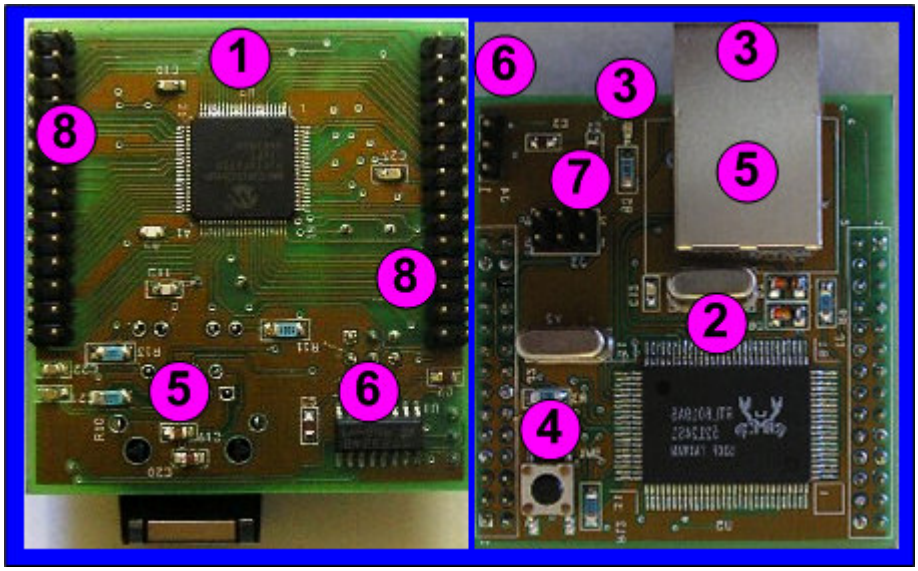


Figura 2 Placa PME-10

Os principais componentes da PME-10 são:

1. **MICROCONTROLADOR**: PIC18F8720 (*clock* de 20MHz, 128K de memória Flash, 1024 bytes de memória EEPROM, 3840 bytes de memória RAM e 68 portas de I/O programáveis das quais 16 podem ser configuradas como entradas analógicas de 10 bits) programado com a pilha TCP/IP Microchip modificada. A pilha TCP/IP usa no máximo 11*Kbytes* de memória Flash e 145 *bytes* de memória RAM (sem incluir os *buffers* de transmissão e recepção), assim há bastante área de memória para aplicações do usuário.
2. **CONTROLADOR *ETHERNET***: A PME-10 usa o controlador de *Ethernet* Realtek RTL8019AS (10Mbps) .
3. **LEDs de indicação**:
   a. Vermelho: indica que a pilha TCP/IP está sendo executada. É programável pelo usuário. No software fornecido o mesmo pisca a cada 1 segundo.
   b. Amarelo: Indica que a conexão Ethernet está ativa (RJ-45).
   c. Verde: indica que a placa está transmitindo ou recebendo um pacote de dados (RJ-45).
4. **CHAVE PUSHBUTTON**: serve para re-inicialização (RESET) do programa pois está conectada ao pino MCLR do microcontrolador.
5. **CONECTOR RJ-45 (10 Base T)**: Provê conectividade Ethernet padrão.

6. **CONECTOR TIPO MOLEX SERIAL RS232**: Provê os sinais Tx, Rx e GND para transmissão serial assíncrona proveniente do integrado ST232ABD.
7. **CONECTOR ICSP (*In Circuit Serial Programming*)**: Permite a placa PME-10 ser conectada a interface de programação para gravação da memória Flash do microcontrolador.
8. **CONECTORES DE EXPANSÃO**: Provê acesso a maior parte dos pinos do microcontrolador PIC18F8720. A PME-10 pode ser alimentada diretamente por este conector (+ 5V nos pinos 13 e 14 de J2, GND no pino 16 de J2).

A placa PME-10 é projetada para executar a pilha TCP/IP fornercida pela Microchip com pequenas modificações. São características da pilha TCP/IP fornecida pela Microchip:

- Inclui MAC, IP, ARP, ICMP, TCP, UDP, HTTP, FTP, DHCP e MPFS.
- Suporte a sockets para protocolos TCP e UDP.
- Suporte para compiladores Microchip C18 e Hi Tech PICC18.
- RTOS Independente.
- Software modular.

3.0 Conector para Placas de Expansão

A PME-10 possui dois conectores macho de 2X14 pinos. Eles contêm a maioria das portas de E/S da CPU PIC18F8720. Os pinos são os seguintes:

Conector de Expansão J1		Conector de Expansão J2	
Pino	**Nome**	**Pino**	**Nome**
1	RC4	1	RF7
2	RC3	2	RF6
3	RC2	3	MCLR
4	RJ7	4	RG4
5	RJ6	5	RG2
6	RJ5	6	RG3
7	RJ4	7	RG0
8	RC7	8	RG1
9	RC6	9	RH2
10	RC0	10	RH3
11	RC1	11	RH0
12	RA4	12	RH1
13	RA5	13	+5V
14	VSS	14	+5V
15	RA0	15	RJ0
16	RA1	16	TERRA
17	RA2	17	RJ2
18	RA3	18	RJ1
19	RA2	19	RB0
20	RF1	20	RJ3
21	RH4	21	RB2
22	RH5	22	RB1
23	RH6	23	RB4
24	RH7	24	RB3
25	RF2	25	RB6
26	RF3	26	RB5
27	RF4	27	RC5
28	RF5	28	RB7

3.1 A PME-10 como Placa Filha

A PME-10 pode ser usada para adicionar capacidades Ethernet para qualquer sistema usando-a como placa filha. A placa que receberá

a PME-10 como filha necessita provê dois conectores fêmeas 2X14 com espaçamento entre os pinos de 2,54 mm.

3.2 A PME-10 com Placa Filha

A PME-10 pode ser usada como um módulo microcontrolado totalmente funcional. Os conectores de expansão podem adicionar funcionalidades das placas filhas. A Figura mostra uma placa filha (FD-01) de expansão conectada a PME-10.

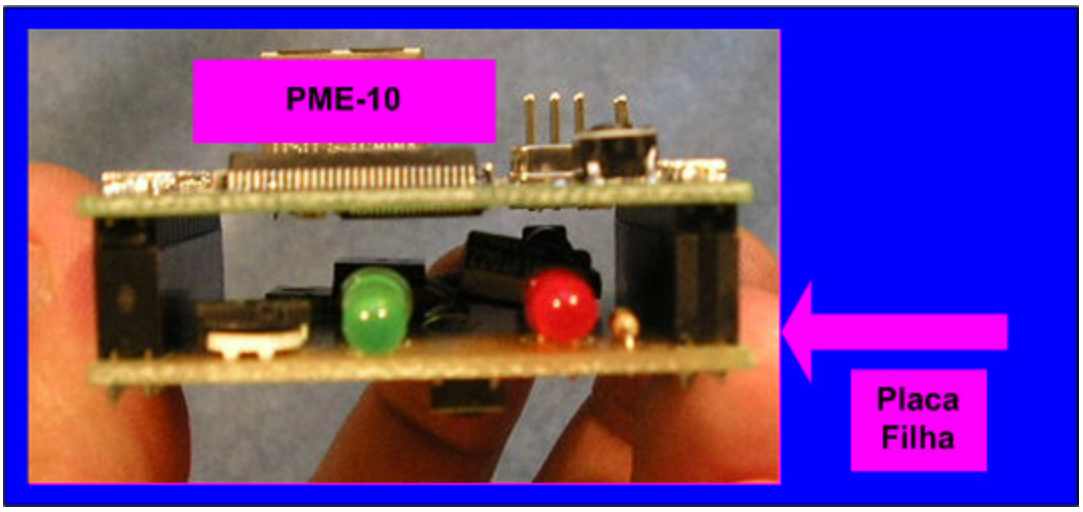


Figura 3 Placa PME-10 conectada a Placa Filha FD-01

4. Interfaces

4.0 Ethernet

A PME-10 possue uma porta Ethernet 10Mbps. O conector RJ-45 atende os requisitos da IEEE 802.3 e requisitos mecânicos FCC. O conector RJ-45 possui dois LEDs. O LED amarelo para indicação de *link* e o LED verde para indicação de atividade de envio ou recepção de dados.

4.1 RS232

A PME-10 possui uma interface USART. Os sinais USART são disponíveis via conector tipo Molex de 3 pinos.

4.2 Conector ICSP

A PME-10 tem um conector ICSP (*In Circuit Serial Programming*) . Este conector possibilita o microcontrolador PIC 18F8720 ser programado e o software carregado ser depurado (caso esteja se usando o ICD).

4.3 Pinos Analógicos e Digitais de E/S

A placa PME-10 possui 52 pinos disponíveis para uso como entradas ou saídas. Entradas e saídas digitais podem estão na faixa de 0 a 5V. Entradas são tolerantes a 3V e as saídas podem ser tolerantes a 3V adicionado-se um resistor em série (assumindo que as entradas de 3V possuem diodos clamping).

A PME-10 pode ser configurada para ter entre um a dezesseis canais analógicos. Cada canal tem resolução de 10 bits. Consulte o manual do microcontrolador PIC 18F8720 para maiores detalhes.

5. Especificações PME-10

Item	Símbolo	Mínima	Típica	Máxima	Unidade
Temperatura de operação	Top	0		70	°C
Temperatura de armazenamento	Tst	-65		140	°C
Tensão DC de alimentação	Vdd	4,75	5	5,25	V
Típica corrente de operação a 20MHz (Vdd=5V)	Idd		80		mA

Pinos de I/O

Item	Símbolo	Condição	Min.	Típ.	Máx.	Unidade
Tensão de entrada TTL nível lógico baixo	V_{IL}		0		0,75	V
Tensão de entrada *Schmitt Trigger* nível lógico baixo	V_{IL}		0		1	V
Tensão de entrada TTL nível lógico alto	V_{IH}		2,05		5	V
Tensão de entrada *Schmitt Trigger* nível lógico alto	V_{IH}		4		5	V
Tensão de saída nível lógico baixo	V_{OL}	I_{OL}=8,5mA			0,6	V
Tensão de saída nível lógico baixo	V_{OH}	I_{OH}=4,3mA	4,3			V
Carga capacitiva	C_{IO}			50		pF

Para maiores detalhes sobre os pinos do PIC18F8720 consulte o respectivo DataSheet.

6. Conectando a Placa PME-10 pela Primeira Vez

6.0 Precauções

A placa PME-10 fornecida com seu kit foi projetada para demonstrar as possibilidades de uso de rede Ethernet ou Internet com o microcontrolador PIC18F8720 da Microchip. Como em qualquer sistema experimental, contudo, algumas precauções devem ser tomadas antes de qualquer novo desenvolvimento.

Sempre que um novo *hardware* ou *software* é adicionado a um sistema em rede, é sempre aconselhável criar uma rede de testes separada da LAN (Local Area *Network*) de uso cotidiano. Isto permite testar o novo sistema em um ambiente controlado e minimizar as possibilidades de interferência na rede de um novo sistema. As maiores fontes de interferência são:

- **ENDEREÇAMENTO.** Cada dispositivo na rede deve ter um único endereço IP. Se o protocolo DHCP (*Dynamic Host Configuration Protocol*) for usado, a PME-10 automaticamente será configurada com um endereço IP. Um endereço fixo deve ser fornecido se o protocolo DHCP não for usado.
- **TRÁFEGO.** Enquanto o controlador *Ethernet* filtra mensagens não endereçadas a PME-10, uma rede *Ethernet* com muitas mensagens *broadcast* pode carregar demasiadamente o microcontrolador PIC18F8720.
- **SEGURANÇA DOS DADOS.** Embora não seja comum que a adição de um dispositivo simples irá comprometer a integridade dos dados ou privacidade de informações, é sempre uma boa idéia executar testes intensivamente com novos sistemas antes de adicioná-los a uma rede.

6.1 Conectando a PME-10 com um Host Local sem DHCP

Embora a PME-10 esteja apta a trabalhar com endereço IP dinâmico (DHCP), você pode ter razões para não fazer isto. Além do que mencionaremos, pode haver outras razões para isto:

- Você pode não ter uma rede *Ethernet* disponível para testes.
- Sua rede não faz uso do protocolo DHCP, de modo que a placa PME-10 deve ser configurada com um endereço IP antes de ser conectada a rede.

- Seu administrador de rede proibe você de colocar a PME-10 na rede para testes.

A PME-10 pode se comunicar diretamente com um computador pessoal quando ambos estiverem configurados corretamente. Esta configuração é conhecida como sistema *host* local. Um sistema para ser um *host* local deve possuir os seguintes requisitos básicos:

- Um computador com interface *Ethernet* com conector RJ-45.
- Sistema operando com pilha TCP/IP.
- Qualquer navegador *Internet* suportando protocolo HTTP 1.0 ou superior.

A placa PME-10 vem configurada de fábrica com endereço IP 10.0.0.101 e máscara de rede 255.255.255.0. Nesta configuração o protocolo DHCP não está habilitado.

Nesta configuração deve-se usar um cabo *crossover* (incluso no *kit*) para interligação da placa PME-10 com um computador pessoal. Para muitos desenvolvedores de sistemas esta opção é a mais indicada para acessar páginas Web que estão gravadas na memória Flash do microcontrolador PIC18F8720.

Na temos o sistema montado. Siga as seguintes etapas para montar o sistema:

Figura 4Conexão entre a PME-10 e um computador Host usando cabo crossover

1. Energize a placa PME-10 através de um dos seguintes modos:
   - Placa PME-10 mais placa de expansão fonte FD-01 e transformador 12VDC, ou
   - Energize a placa com 5VDC no conector de expansão pinos (5VDC nos pinos 13 e 14 do conector J2 e GND no pino 16 do conector J2).
2. Verifique que o LED D3 (vermelho) na PME-10 pisca a cada intervalo de 1 segundo.
3. Conecte o cabo *crossover* entre a PME-10 e a placa de rede Ethernet do computador que contém o navegador Web.
4. Configure o computador do seguinte modo:
   - Endereço IP: 10.10.5.X (aonde X deve ser diferente de 15)
   - Máscara de rede: 255.255.255.0

O apêndice A contém informações de como configurar o Sistema Operacional Windows XP para endereço IP e máscara de rede.

5. No computador, navegador Web, na barra de endereços digite http://10.10.5.15 e tecle <ENTER>.
6. A página Web (Figura 5) deverá ser carregada na tela de seu computador.

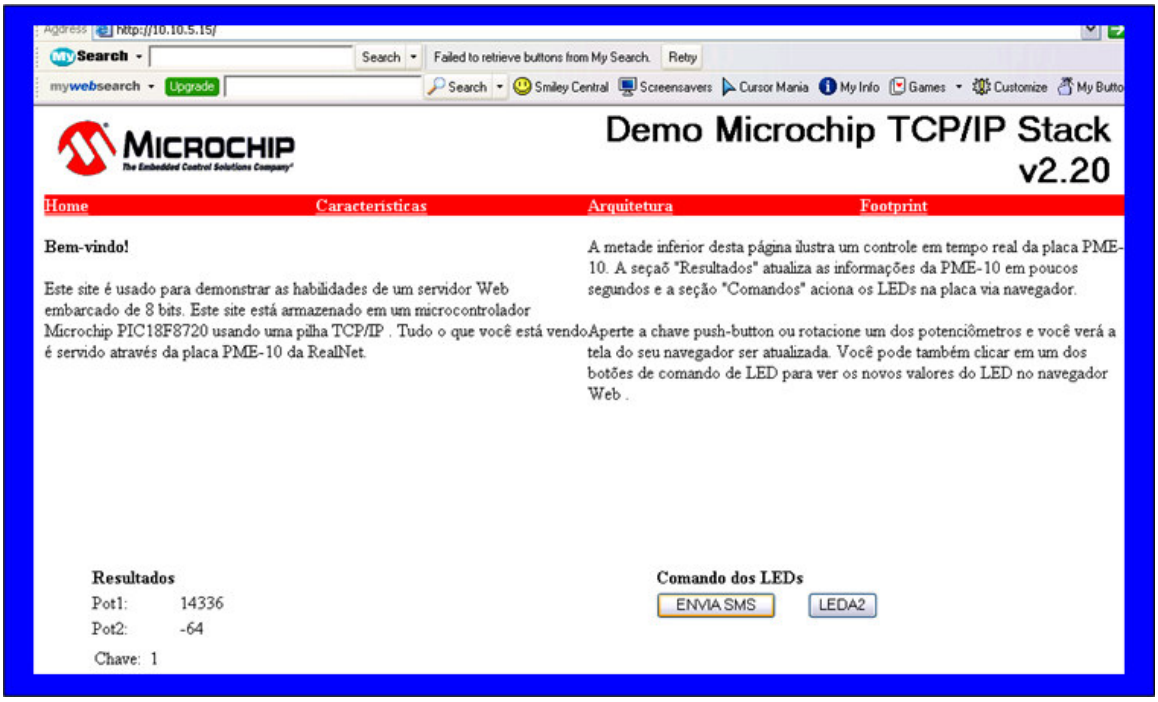

http://10.10.5.15/
Search
Failed to retrieve buttons from My Search.
Retry
Upgrade
Smiley Central
Screensavers
Cursor Mania
My Info
Games
Customize
MICROCHIP
Demo Microchip TCP/IP Stack
v2.20
Home
Caracteristicas
Arquitetura
Footprint
Bem-vindo!
Este site é usado para demonstrar as habilidades de um servidor Web embarcado de 8 bits. Este site está armazenado em um microcontrolador Microchip PIC18F8720 usando uma pilha TCP/IP . Tudo o que você está vendo é servido através da placa PME-10 da RealNet.
A metade inferior desta página ilustra um controle em tempo real da placa PME-10. A seçaõ "Resultados" atualiza as informações da PME-10 em poucos segundos e a seção "Comandos" aciona os LEDs na placa via navegador.
Aperte a chave push-button ou rotacione um dos potenciômetros e você verá a tela do seu navegador ser atualizada. Você pode também clicar em um dos botões de comando de LED para ver os novos valores do LED no navegador Web .
Resultados
Pot1: 14336
Pot2: -64
Chave: 1
Comando dos LEDs
ENVIA SMS
LEDA2

Figura 5 Página Web PME-10

Caso você esteja usando a placa PME-10 juntamente com a placa filha fonte-demonstração FD-01 os seguintes recursos podem ser verificados:

- Varie o potenciômetro P1 verificando que o valor do Pot1 também varia na página do seu navegador Web.
- No navegador Web, clique no botão LED1 verificando que o estado (ligado e desligado) do LED1 (verde) varia na placa FD-01 também.
- Aperte a tecla push-button na FD-01 e verifique que o estado da Chave (ligada ou desligada) varia na página do navegador Web também.

6.2 Alterando o endereço IP sem usar o DHCP

A 2EI possui endereço MAC próprio de acordo com o "Organizationally Unique Identifier (OUI)" do IEEE. Os octetos da Microchip (00.04.A3) deverão ser substituídos pelos octetos da 2EI (00.16.A9).
Esta alteração deverá ser feita no arquivo StackTsk.h , nas linhas 283 a 288.
#define MY_DEFAULT_MAC_BYTE1 (0x00)
#define MY_DEFAULT_MAC_BYTE2 (0x16)
#define MY_DEFAULT_MAC_BYTE3 (0xA9)

Continuam como estão na placa, conforme etiqueta adesiva:
#define MY_DEFAULT_MAC_BYTE4 (0x??)
#define MY_DEFAULT_MAC_BYTE5 (0x??)
#define MY_DEFAULT_MAC_BYTE6 (0x??)

Para mudar o endereço IP da placa PME-10 deve-se fazer o uso do cabo de programação que acompanha o kit de desenvolvimento. Os seguintes procedimentos devem ser executados para fazer a mudança do endereço IP da PME-10.

1. Instalando e Executando o MPLAB IDE

É aconselhável aceitar o nome das pastas sugeridas nos programas de instalação. Embora os arquivos sejam instalados corretamente quando você selecionar outros destinos, eles podem não estar disponíveis para o seu ambiente de desenvolvimento se você decidir criar um novo projeto usando a pilha TCP/IP.

i. O primeiro passo é criar um projeto e uma área de trabalho no sistema de ambiente integrado MPLAB IDE. Um projeto contém os arquivos necessários para construir uma aplicação (códigos fontes, etc.). Uma área de trabalho contém um ou mais projetos e informações dos dispositivos selecionados, ferramentas de depuração, localização das

janelas abertas e outras configurações IDE. O MPLAB IDE contém um Project Wizard para ajudar a criar novos projetos. Utilizaremos projetos já prontos. Copie a pasta Stack_2EI que se encontra no CD para o disco rígido do seu computador. Esta pasta contém vários projetos já criados.

ii. Verifique que a pasta Stack_2EI contém uma pasta Source que contém as pastas WebPages e SmallWebPages.

Arquivos copiados do CD são somente de leitura; você necessitará trocar os atributos dos arquivos que necessitam ser editados.

iii. Instale o ambiente de desenvolvimento MPLAB IDE.

iv. Instale o compilador MCC18.

v. Execute o programa MPLAB IDE.

vi. Verifique a seleção de linguagens no aplicativo MPLAB IDE . Na barra de menu selecione Project\Set Language Tool Locations. Na janela Set language Tool Locations selecione opção Microchip C18 Toolsuite (clique em +). Clique no + de Executables. Clique no + de Default Search Paths & Directories. Verifique:

- Executables
    i. MPASM Assembler (mpasmwin.exe) c:\Arquivos de programas\MPLAB IDE\MCHIP_Tools\ mpasmwin.exe
    ii. MPLAB C18 C Compiler (mcc18.exe) c:\mcc18\bin\c18demo.exe
    iii. MPLINK Object Linker (mplink.exe) c:\mcc18\bin\mplink.exe
- Default Search Path & Directories
    i. Output Directory, $(BINDIR)
    ii. Intermediate s Directory,$(TMPDIR)
    iii. Include Search Path,$(INCDIR)
    iv. Library Search path,$(LIBDIR) c:\mcc18\lib
    v. Linker_Script Search Path,$(LKDIR) c:\mcc18\lib

vii. No MPLAB IDE, selecione na barra de menu Project\Open . Selecione o projeto MpNICPG.pjt da pasta Stack_2EI do seu disco rígido.

viii. No MPLAB IDE, selecione na barra de menu View\Project. Em Linker Script selecione 18f8720.lkr (c:\mcc18\lkr).

ix. Vá ao menu Project e selecione Build All. O projeto será compilado e linkado. Um novo arquivo mpnicpg.hex está pronto para ser transferido para a PME-10.

licação é MpNICPG.pjt. Ele usa NIC e memória de programa interna do microcontrolador para armazenamento do Web Site.

2. Alterando o Endereço IP e Máscara de Rede

Abra o arquivo StackTsk.h usando o comando da barra de menu File\Open. As definições abaixo contêm o endereço IP:

```
#define MY_DEFAULT_IP_ADDR_BYTE1 (10)
#define MY_DEFAULT_IP_ADDR_BYTE2 (10)
#define MY_DEFAULT_IP_ADDR_BYTE3 (5)
#define MY_DEFAULT_IP_ADDR_BYTE4 (15)
```

Obs: significa endereço IP 10.10.5.15

```
#define MY_DEFAULT_MASK_BYTE (0xff)
#define MY_DEFAULT_MASK_BYTE (0xff)
#define MY_DEFAULT_MASK_BYTE (0xff)
#define MY_DEFAULT_MASK_BYTE (0x00)
```

Obs: significa máscara 255.255.255.0

```
#define MY_DEFAULT_MAC_BYTE1 (0x00)
#define MY_DEFAULT_MAC_BYTE2 (0x16)
#define MY_DEFAULT_MAC_BYTE3 (0xa9)
#define MY_DEFAULT_MAC_BYTE4 (0x00)
#define MY_DEFAULT_MAC_BYTE5 (0x00)
#define MY_DEFAULT_MAC_BYTE6 (0x00)
```

Significa endereço MAC: 00.16.a9.00.00.00

Os endereços MAC de cada placa devem ser diferentes uns dos outros.

Faça as modificações que julgar necessário e compile o projeto novamente. O novo arquivo mpnicpg.hex gerado conterá as modificações que você efetuou.

2. Instalando e Executando o Programa IC-Prog 1.05D (Somente para PIC18F8720 e PIC18F8621)

O programa IC-Prog é um programador de protótipos desenvolvido por Bonny Gizen (www.ic-prog.com). Para instalar o IC-Prog siga os seguintes passos:

1. Copie a pasta ICPROG que se encontra no CD para o disco rígido do seu computador.
2. Execute o programa ICPROG a partir do disco rígido do seu computador. Selecione a serial em que você colocará o cabo de programação na janela "Hardware settings". O programador selecionado deve ser JDM Programmer. Se você estiver usando o Windows XP ou NT selecione Windows API, caso contrário selecione Direct IO. A opção Retardar I/O pode estar com o valor mínimo. Caso a programação da placa não tenha sido efetuada corretamente selecione um valor progressivamente maior de Retardar I/O até que a programação ocorra sem erros.

A interface de programação é de custo muito baixo, mas não é capaz de verificar se a programação foi feita de forma correta. Caso o led D3 não fique piscando imediatamente após voltar a chave de programação para a posição de operação, indica que o programa não foi gravado corretamente.

3. Na barra de menu selecione Settings\Options. Clique na aba Language. Selecione a linguagem *Portuguese*.
4. Na barra de menu selecione Configuração\Opções. Clique na aba Programando. A opção "Verifique depois da programação " não deve estar selecionada.
5. Na barra de menu selecione Configuração\Opções. Clique na aba "Diversos". Em "Processo Prioritário" selecione "Alto".
6. No aplicativo ICPROG, na barra de menus, selecione Configuração\ Dispositivo\Microchip PIC\PIC18F8720.
7. Na barra de menus, selecione Arquivo\Abrir e carregue o arquivo HEX mpniccpg que se encontra na pasta C:\Stack_E2I\Source.

8. No programa IC-Prog, os registradores de configuração devem estar carregados com os seguintes valores:

- Config1 - 0x2200
- Config2 - 0x0004
- Config3 - 0x0183
- Config4 - 0x0081
- Config5 - 0xC0FF
- Config6 - 0xE0FF
- Config7 - 0x40FF
- Valor ID - 0x0600

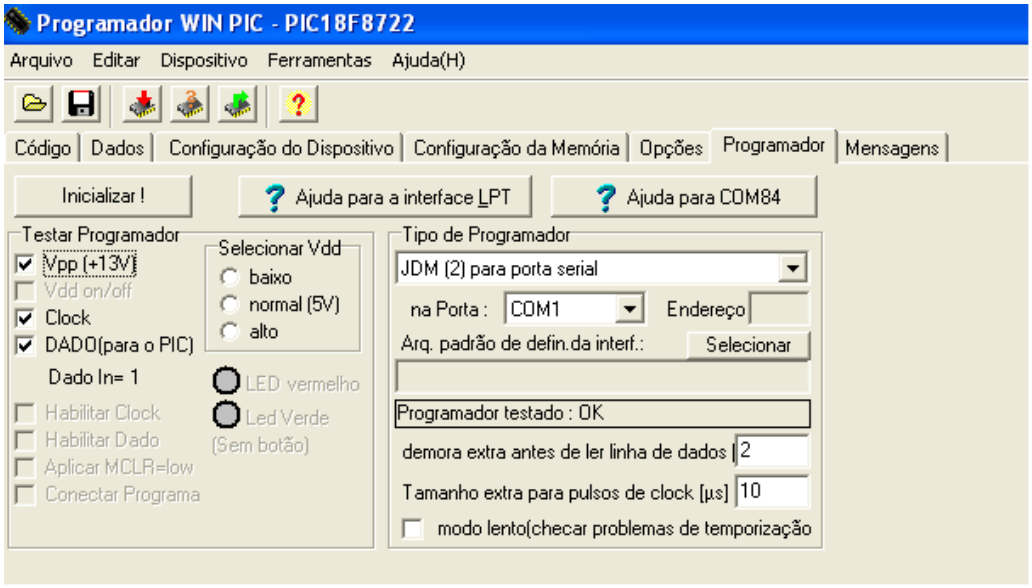
9. Alimente a placa PME-10 com 5 volts.
10. Conecte o cabo de programação na saída serial de seu computador e na placa PME-10.
11. Na interface de programação pressione as chaves de modo que elas habilitem a programação do microcontrolador da PME-10. (▄ - Cursor da chave posição de programar , ▀ - Cursor da chave posição de não programar).
12. Na barra de menu selecione Comando e depois Programar Tudo. O LED da interface de programação deverá ficar ligado.
13. Espere a finalização da programação.
14. Pressione a chave de programação de modo que ela libere ao sinal de RESET do microcontrolador.
15. Execute os procedimentos do item 7.2.

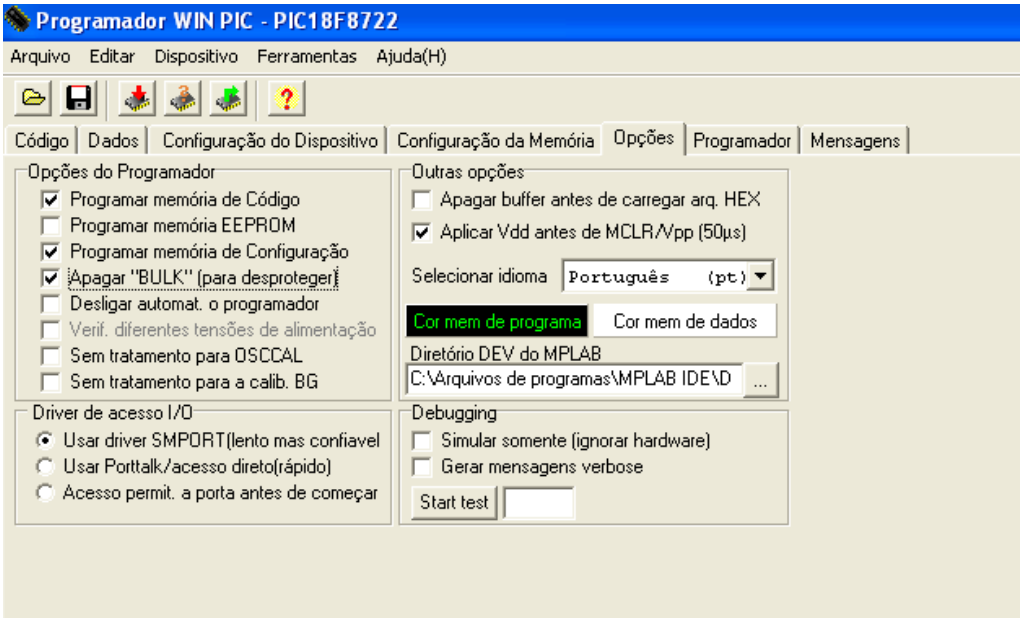
Durante a programação, usando a IP10, jamais conecte outro terra a PME-10 sem ser os da interface de programação. Por exemplo, se sua aplicação estiver usando a interface serial USART RS232 você deverá desconectar o cabo serial sempre durante o processo de gravação. Usando outras interfaces de programação, como o MPLAB ICD, este procedimento não é necessário.

3. Instalando o Programador WINPIC (PIC18F8722)

Instale o programa WinPicSetup que se encontra na pasta WinPIC do CDROM. Substitua o arquivo "devices" pelo o que se encontra no CD-ROM na pasta WinPIC. O arquivo "devices" que se encontra no CDROM possui os parâmetros para gravação do PIC18F8722.

Configure o WinPIC de acordo com as telas a seguir:

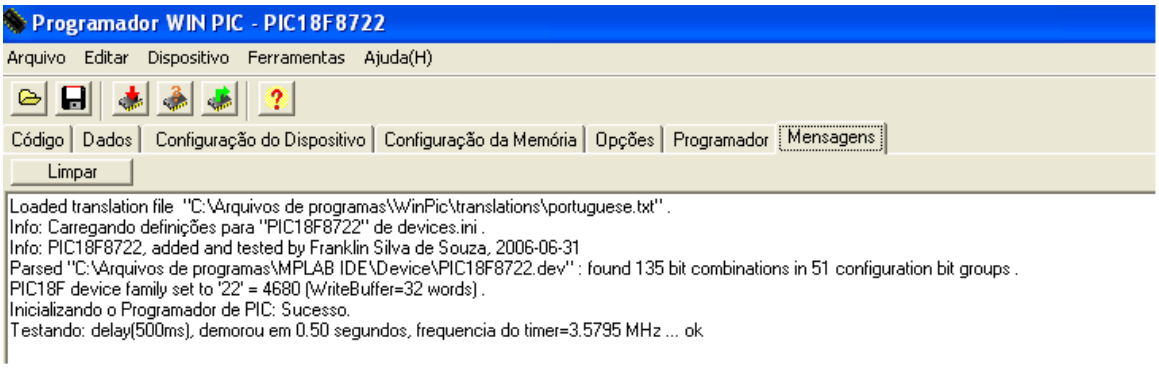



Programador WIN PIC - PIC18F8722
Arquivo Editar Dispositivo Ferramentas Ajuda(H)
Código Dados Configuração do Dispositivo Configuração da Memória Opções Programador Mensagens
Opções do Programador
Programar memória de Código
Programar memória EEPROM
Programar memória de Configuração
Apagar "BULK" (para desproteger)
Desligar automat. o programador
Verif. diferentes tensões de alimentação
Sem tratamento para OSCCAL
Sem tratamento para a calib. BG
Outras opções
Apagar buffer antes de carregar arq. HEX
Aplicar Vdd antes de MCLR/Vpp (50µs)
Selecionar idioma Português (pt)
Cor mem de programa
Cor mem de dados
Diretório DEV do MPLAB
C:\Arquivos de programas\MPLAB IDE\D
Driver de acesso I/O
Usar driver SMPORT(lento mas confiavel
Usar Porttalk/acesso direto(rápido)
Acesso permit. a porta antes de começar
Debugging
Simular somente (ignorar hardware)
Gerar mensagens verbose
Start test

Vá no menu "Arquivo" , opção "Abrir", e carregue o arquivo "mpnicpg.hex" que se encontra na pasta Pilhas\PME10\v1.15\Source.

Selecione a aba mensagens.

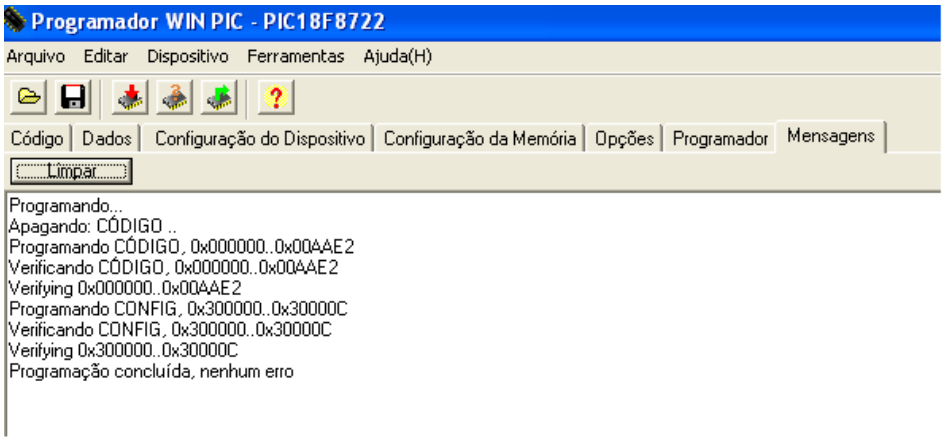
Verifique que WriteBuffers=32 words, caso isto não aconteça, consulte a documentação do WINPIC. Uma das alternativas para resolver este problema é executar o WINPIC através da linha de comando "winpic /device=xxxx". Se não conseguir resolver o problema entre em contato com franklin_souza@uol.com.br.


Programador WIN PIC - PIC18F8722
Arquivo Editar Dispositivo Ferramentas Ajuda(H)
Código Dados Configuração do Dispositivo Configuração da Memória Opções Programador Mensagens
Limpar
Loaded translation file "C:\Arquivos de programas\WinPic\translations\portuguese.txt".
Info: Carregando definições para "PIC18F8722" de devices.ini .
Info: PIC18F8722, added and tested by Franklin Silva de Souza, 2006-06-31
Parsed "C:\Arquivos de programas\MPLAB IDE\Device\PIC18F8722.dev" : found 135 bit combinations in 51 configuration bit groups .
PIC18F device family set to '22' = 4680 (WriteBuffer=32 words) .
Inicializando o Programador de PIC: Sucesso.
Testando: delay(500ms), demorou em 0.50 segundos, frequencia do timer=3.5795 MHz ... ok

Verifique na Interface de Programação as chaves "RESET" e "CK/DAD" devem estar pressionadas, ou seja, em baixo.

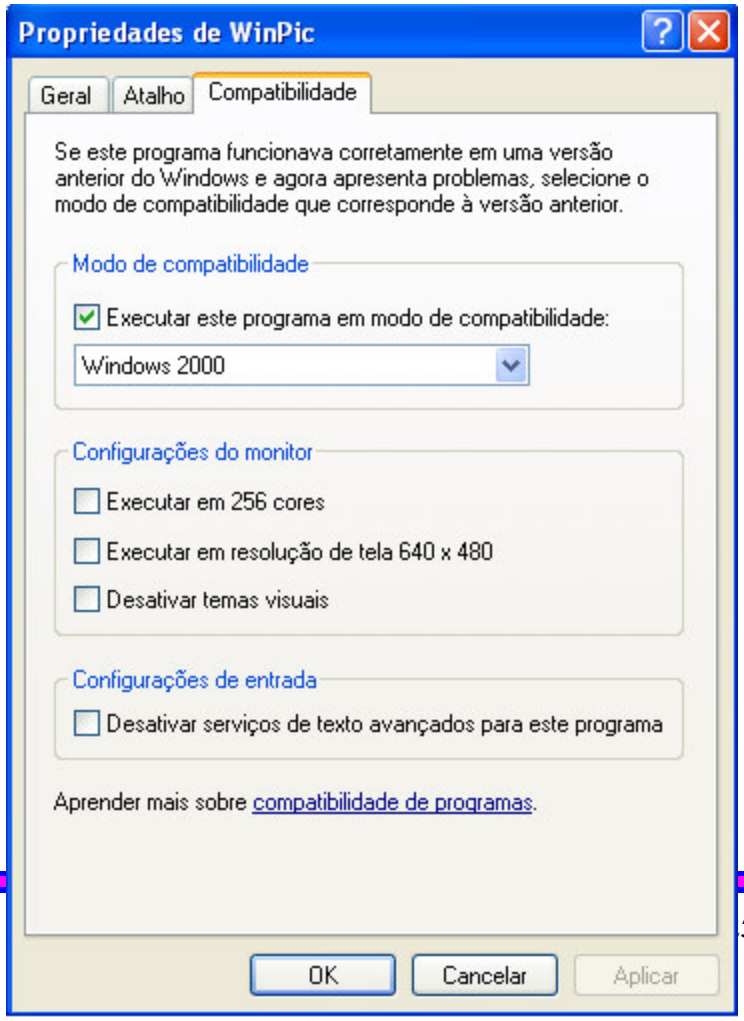
Clique no botão "Programar Dispositivo".

A programação deve ocorrer sem nenhum erro, conforme figura a seguir.



Para execução do programa as chaves da Interface de Programação RESET e CK/DAD não devem estar pressionadas.

Caso ocorram erros durante a programação, sugiro que o programa WinPIC seja executado no modo de compatibilidade Windows 2000, conforme a figura a seguir:


Propriedades de WinPic
Geral
Atalho
Compatibilidade
Se este programa funcionava corretamente em uma versão anterior do Windows e agora apresenta problemas, selecione o modo de compatibilidade que corresponde à versão anterior.
Modo de compatibilidade
Executar este programa em modo de compatibilidade:
Windows 2000
Configurações do monitor
Executar em 256 cores
Executar em resolução de tela 640 x 480
Desativar temas visuais
Configurações de entrada
Desativar serviços de texto avançados para este programa
Aprender mais sobre compatibilidade de programas.
OK
Cancelar
Aplicar

4. Modificando as Páginas Web

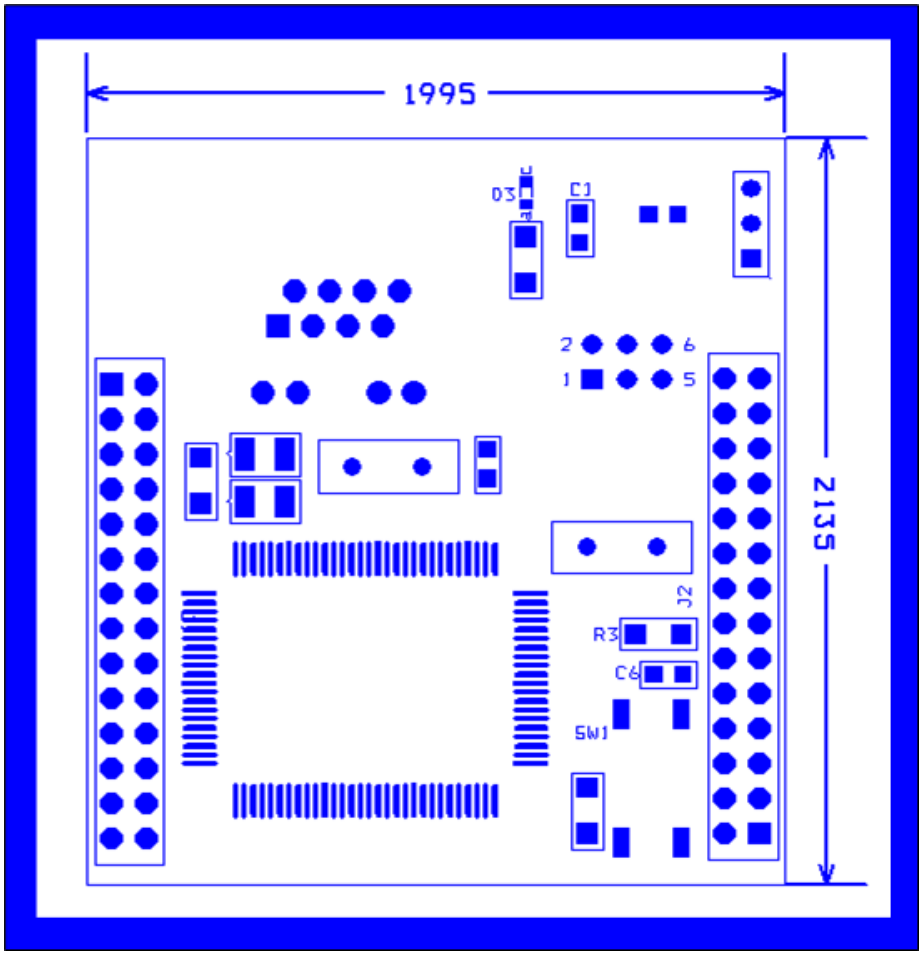
1. Faça as modificações necessárias na página INDEX.HTML ou em outras páginas .HTML que se encontram na pasta C:\Stack2EI\Source\WebPages usando um editor de textos. Você deve ter conhecimentos em HTML para modificar estas páginas ou criar novas páginas Web usando aplicativos como o Microsoft FrontPage.
2. Na pasta C:\PIC18F\Stack2EI\Source execute o comando mpfs /c webpages MPFSImg.c (ou simplesmente execute o arquivo batch imagem.bat) para gerar um novo arquivo .C que contém as imagens das novas páginas geradas.
3. No programa MPLAB IDE , selecione a opção Project\Build All para gerar um novo arquivo ,mpniccpg.hex.
4. utilize o programa IC-Prog e o cabo de programação para gravar o novo arquivo mpniccpg .hex que contém as novas páginas Web.

Não esqueça de verificar a posição do cabo que interliga a interface de programação com a placa PME-10. A etiqueta .INT indica que o conector deve ser colocado na interface de programação pino 1. A etiqueta .PME10 significa que o conector deve ser colocado na placa PME-10

6.3 Usando o DHCP

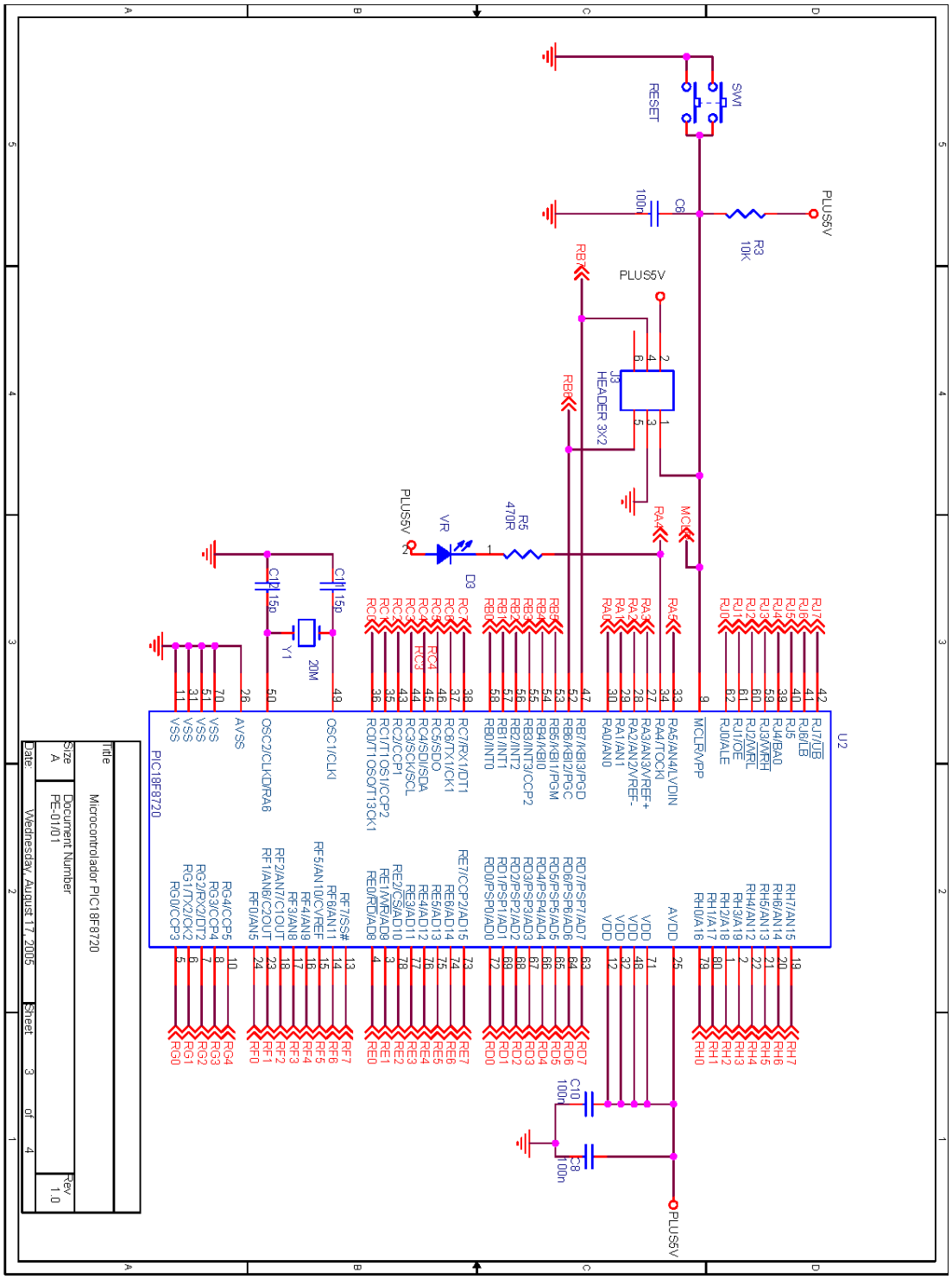
Para uso de endereço IP dinâmico, ou seja, uso do protocolo DHCP é necessário incluir a macro definição "STACK_USE_DHCP" nas "Build Options" de cada arquivo (Apêndice B). Siga os mesmos procedimentos do item 5.2 para gravação da memória Flash do microcontrolador PIC18F8720. Para saber o endereço IP dinâmico é necessário perguntar ao seu administrador de rede ou usar um display de LCD adicionando a macro definição "USE_LCD" nos arquivos que constituem o projeto.

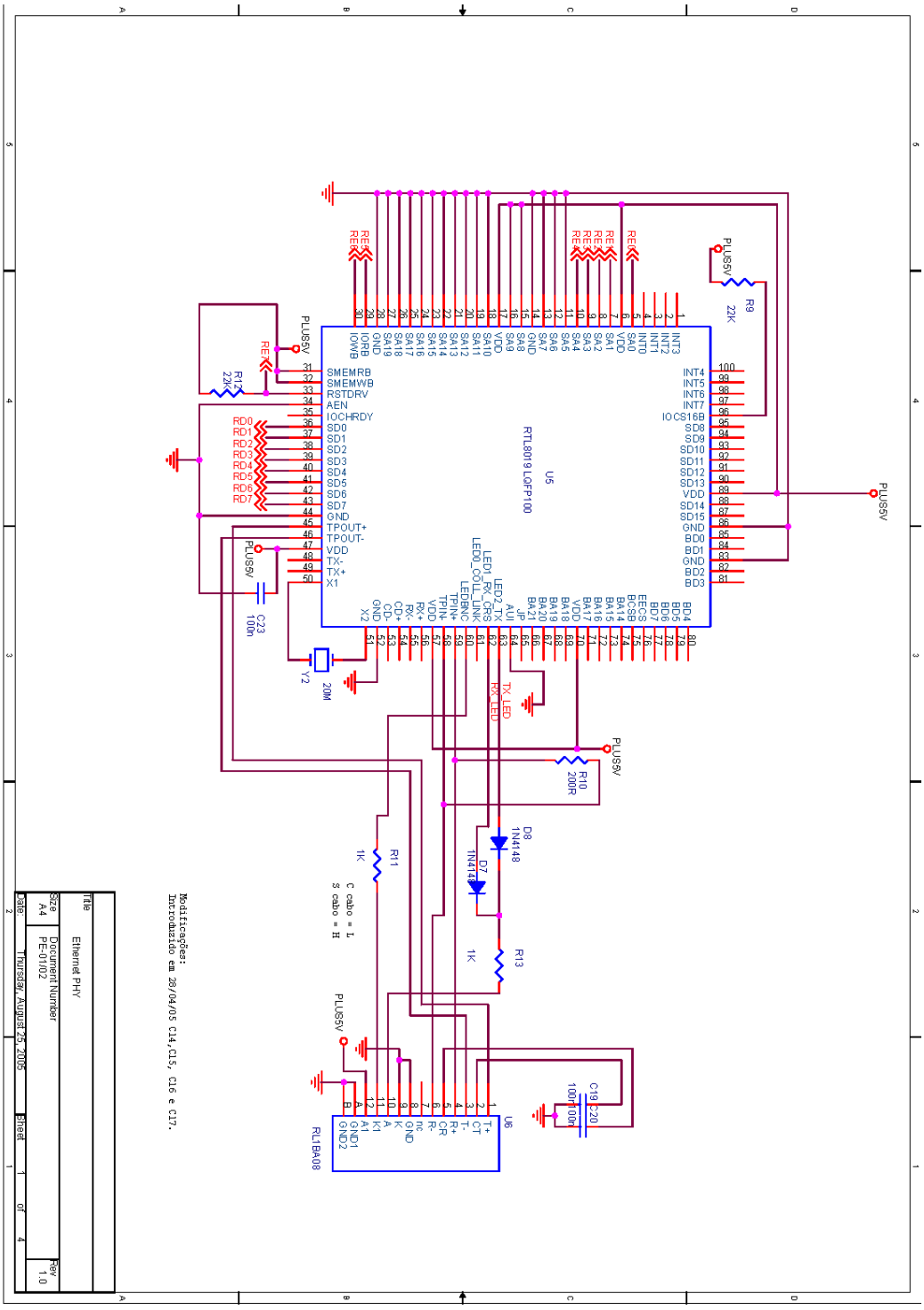
7. Dimensões da PME-10

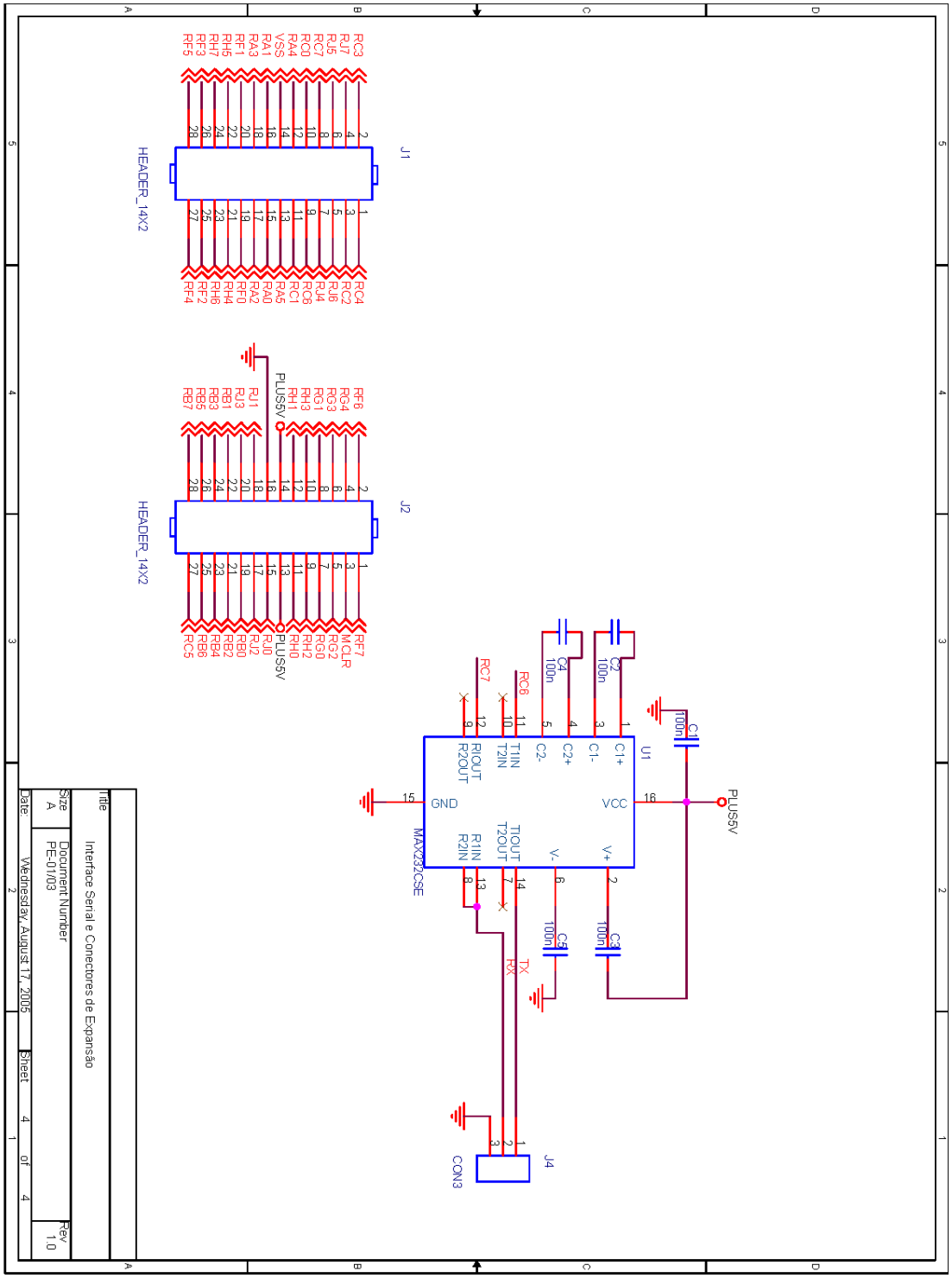


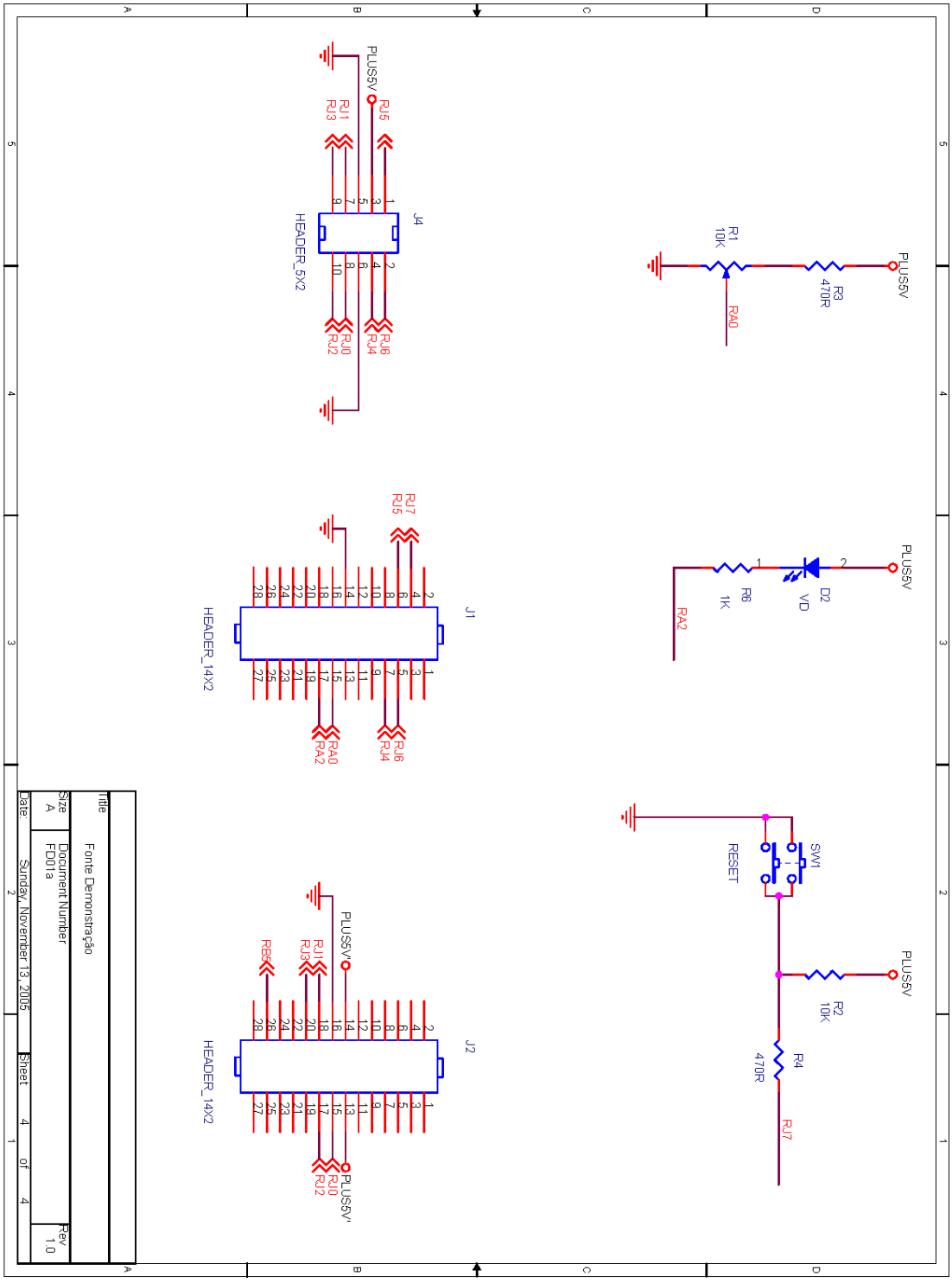
As dimensões estão em mil () de polegada de modo que em centímetros são : 51(1995) X 54(2135) mm.

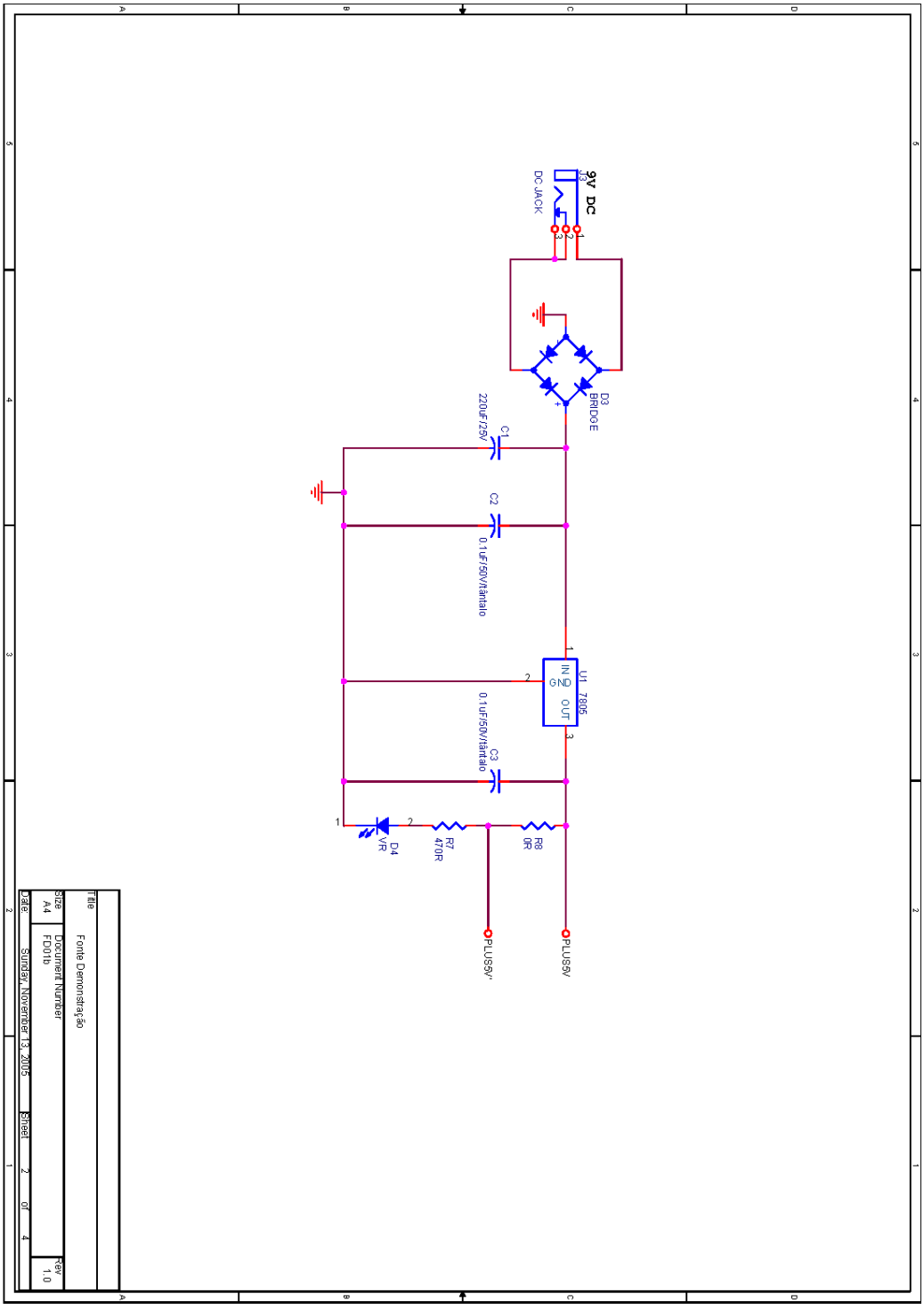
8. Esquemas


SW1
RESET
C6
100n
R3
10K
PLUS5V
J3
HEADER 3X2
VR
R5
470R
D3
C11
15p
C12
15p
Y1
20M
C10
100n
C8
100n
U2
PIC18F8720
Title
Microcontrolador PIC18F8720
Size
A
Document Number
PE-01/01
Rev
1.0
Date
Wednesday, August 17, 2005
Sheet
3
of
4


RTL8019 LQFP100
U5
PLUS5V
R9
22K
R12
22R
C23
100n
Y2
20M
R10
200R
R11
1K
R13
1K
D8
1N4148
D7
1N4148
RL1BA08
U6
C19
100n
C20
100n
Ethernet PHY
Document Number
PE-01102
Thursday, August 25, 2005
Rev
1.0
Modificações:
Incrementado em 28/04/05 C14,C15, C16 e C17.


HEADER_14X2
J1
J2
PLUS5V
MAX232CSE
U1
C1
C2
C3
C4
C5
CON3
J4
GND
VCC
TX
RX
Title
Interface Serial e Conectores de Expansão
Size
A
Document Number
PE-01/03
Rev
1.0
Date
Wednesday, August 17, 2005
Sheet
4
of
4


PLUS5V
RJ1
RJ3
RJ5
J4
HEADER_5X2
RJ0
RJ2
RJ6
RJ4
R1
10K
RA0
R3
470R
RJ7
RJ5
J1
HEADER_14X2
RA0
RA2
RJ6
RJ4
R6
1K
D2
VD
RA2
SW1
RESET
R2
10K
R4
470R
RJ7
RB5
J2
RJ0
RJ2
PLUS5V'
Title
Fonte Demonstração
Size
A
Document Number
FD01a
Rev
1.0
Date:
Sunday, November 13, 2005
Sheet
4
of
4


9V DC
DC JACK
D3
BRIDGE
220uF/25V
C1
C2
0.1uF/50V/tântalo
U1
7805
IN
GND
OUT
0.1uF/50V/tântalo
C3
D4
VR
R7
470R
R8
0R
PLUS5V
PLUS5V
Title
Fonte Demonstração
Size
A4
Document Number
FD01b
Rev
1.0
Date:
Sunday, November 13, 2005
Sheet
2
of
4

9. Configuração do PIC18F8720

As tabelas a seguir auxiliam configuração do PIC18F8720.

Tabela 1 Bits de Configuração PIC18F8720

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Padrão
300001h	CONFIG1H	-	-	OSCEN	-	-	FOSC2	FOSC1	FOSC0	0010 0111
300002h	CONFIG2L	-	-	-	-	BORV1	BORV0	BODEN	PWRTEN	0000 1111
300003h	CONFIG2H	-	-	-	-	WDTPS2	WDTPS1	WDTPS0	WDTEN	0000 1111
300004h	CONFIG3L	WAIT	-	-	-	-	-	PM1	PM0	1000 0011
300005h	CONFIG3H	-	-	-	-	-	-	-	CCP2MX	0000 0001
300006h	CONFIG4L	DEBUG	-	-	-	-	LPV	-	STVREN	1000 0101
300008h	CONFIG5L	CP7	CP6	CP5	CP4	CP3	CP2	CP1	CP0	1111 1111
300009h	CONFIG5H	CPD	CPB	-	-	-	-	-	-	1100 0000
30000Ah	CONFIG6L	WRT7	WRT6	WRT5	WRT4	WRT3	WRT2	WRT1	WRT0	1111 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	-	-	-	-	-	1110 0000
30000Ch	CONFIG7L	EBTR7	EBTR6	EBTR5	EBTR4	EBTR3	EBTR2	EBTR1	EBTR0	1111 1111
30000Dh	CONFIG7H	-	EBTRB	-	-	-	-	-	-	0100 0000
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	Tab.
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	Tab.

Tabela 2 Bits de CONFIG1, CONFIG2 e CONFIG3

Nome do Bit	Words	Descrição
OSCEN	CONFIG1H	Opção Low power System Clock (Timer1) Enable Bit 1 = Desabilitado 0 = Habilitado
FOSC2:FOSC0	CONFIG1H	Seleção dos bits do oscilador 111 = RC com OSC2 configurado como RA6 110 = HS com PLL habilitado 101 = EC com OSC2 configurado como RA6 100 = RC com OSC2 configurado como “dividido por 4 clock output” 011 = RC 010 = HS 001 = XT 000 = LP
BORV1:BORV0	CONFIG2L	Bits Brown-out reset Voltage 11 = VBOR para 2,0V 10 = VBOR para 2,7V 01 = VBOR para 4,2V 00 = VBOR para 4,5V
BOREN	CONFIG2L	Bit Brown-out Reset Enable 1 = Habilitado 0 = Desabilitado
PWRTEN	CONFIG2L	Bit Power-up Timer Enable 1 = Desabilitado 0 = Habilitado
WDTPS2:WDTPS0	CONFIG2H	Bits WatchDog Postscaler Select 111 = 1:128 110 = 1:64 101 = 1:32 100 = 1:16 011 = 1:8 010 = 1:4 001 = 1:2 000 = 1:1
WDTEN	CONFIG2H	Bit Watchdog Timer Enable 1 = Habilitado 0 = Desabilitado
WAIT	CONFIG3L	Bit External Bus Data Wait Enable 1 = Seleção indisponível 0 = Seleção determinada pelos bits WAIT1:WAIT0 do registrador MEMCOM
PM1:PM0	CONFIG3L	Bits Processor Mode Select 11 = microcontrolador 10 = Microprocessador 01 = Microprocessador com modo Boot Block 00 = Microcontrolador extendido

Tabela 3 Bits de CONFIG3, CONFIG4 e CONFIG5

Nome do Bit	Words	Descrição
T10SCMX3	CONFIG3H	Não impl. 18F8720
CCPMX	CONFIG3H	Bit MUXCCP2 1 = Entrada/saída CCP2 multiplexada com RC1 0 = Entrada/saída CCP2 multiplexada com RB3
DEBUG	CONFIG4L	Bit background Debugger Enable 1 = Desabilitado 0 = Habilitado
LVP	CONFIG4L	Bit Low Voltage Programming enable 1 = Habilitado 0 = Desabilitado
STVREN	CONFIG4L	Bit Stack Overflow/Underflow Reset Enable 1 = Stack causará RESET 0 = Stack não causará RESET
CP0	CONFIG5L	Bits Code Protection (Block 0) 1 = Código de memória não protegido 0 = Código de memória protegido
CP1	CONFIG5L	Bits Code Protection (Block 1) 1 = Código de memória não protegido 0 = Código de memória protegido
CP2	CONFIG5L	Bits Code Protection (Block 2) 1 = Código de memória não protegido 0 = Código de memória protegido
CP3	CONFIG5L	Bits Code Protection (Block 3) 1 = Código de memória não protegido 0 = Código de memória protegido
CP4	CONFIG5L	Bits Code Protection (Block 4) 1 = Código de memória não protegido 0 = Código de memória protegido
CP5	CONFIG5L	Bits Code Protection (Block 5) 1 = Código de memória não protegido 0 = Código de memória protegido
CP6	CONFIG5L	Bits Code Protection (Block 6) 1 = Código de memória não protegido 0 = Código de memória protegido
CP7	CONFIG5L	Bits Code Protection (Block 7) 1 = Código de memória não protegido 0 = Código de memória protegido

Tabela 4 Bits de CONFIG 5 e CONFIG6

Nome do Bit	Words	Descrição
CPD	CONFIG5H	Bits Code Protection (Data EEPROM) 1 = Código Data EEPROM não protegido 0 = Código Data EEPROM protegido
CPB	CONFIG5H	Bits Code Protection (Bloco de Boot) 1 = Código do bloco de boot não protegido 0 = Código do bloco de boot protegido
WRT0	CONFIG6L	Bit Table Write Protection (Bloco 0) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT1	CONFIG6L	Bit Table Write Protection (Bloco 1) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT2	CONFIG6L	Bit Table Write Protection (Bloco 2) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT3	CONFIG6L	Bit Table Write Protection (Bloco 3) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT4	CONFIG6L	Bit Table Write Protection (Bloco 4) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT5	CONFIG6L	Bit Table Write Protection (Bloco 5) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT6	CONFIG6L	Bit Table Write Protection (Bloco 6) 1 = Código de memória não protegido 0 = Código de memória protegido
WRT7	CONFIG6L	Bit Table Write Protection (Bloco 7) 1 = Código de memória não protegido 0 = Código de memória protegido
WRTD	CONFIG6H	Bit Table Write Protection (Data EEPROM) 1 = Data EEPROM não protegido 0 = Data EEPROM protegido
WRTB	CONFIG6H	Bit Table Write Protection (Bloco de boot) 1 = Bloco de Boot não protegido 0 = Bloco de boot protegido
WRTC	CONFIG6H	Bit Table Write Protection (Registradores de configuração) 1 = Registradores de configuração não protegidos 0 = Registradores de configuração protegidos

Tabela 5 Bits de CONFIG7 e DEVID

Nome do Bit	Words	Descrição
EBTR0	CONFIG7L	Bit Table Read Protection (Bloco 0) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR1	CONFIG7L	Bit Table Read Protection (Bloco 1) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR2	CONFIG7L	Bit Table Read Protection (Bloco 2) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR3	CONFIG7L	Bit Table Read Protection (Bloco 3) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR4	CONFIG7L	Bit Table Read Protection (Bloco 4) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR5	CONFIG7L	Bit Table Read Protection (Bloco 5) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR6	CONFIG7L	Bit Table Read Protection (Bloco 6) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTR7	CONFIG7L	Bit Table Read Protection (Bloco 7) 1 = Código de memória não protegido da tabela de leitura executada em outros blocos 0 = Código de memória protegido da tabela de leitura executada em outros blocos
EBTRB	CONFIG7H	Bit Table Read Protection (Bloco de boot) 1 = Código de boot não protegido da tabela de leitura executada em outros blocos 0 = Código de boot protegido da tabela de leitura executada em outros blocos
DEV10:DEV3	DEVID2	Bits Device ID Estes bits são usados com os bits DEV2:DEV0 no registrador DEVID1 para identificar números de componentes.
DEV2:DEV0	DEVID1	Bits Device ID Estes bits são usados com os bits DEV10:DEV3 no registrador DEVID1 para identificar números de componentes
REV4:REV0	DEVID1	Estes bits são usados para identificar revisões de dispositivos

Tabela 6 Configuração gravada na entrega do kit

CONFIGURAÇÃO PIC18F8720			
CONFIG1L	0x0000 0x0000	0x00	0x2200
CONFIG1H	0x0010 0x0010	0x22	
CONFIG2L	0x0000 0x0100	0x04	0x0004
CONFIG2H	0x0000 0x0000	0x00	
CONFIG3L	0x1000 0x0011	0x83	0x0183
CONFIG3H	0x0000 0x0001	0x01	
CONFIG4L	0x1000 0x0001	0x81	0x0081
CONFIG4H	0x0000 0x0000	0x00	
CONFIG5L	0x1111 0x1111	0xFF	0xC0FF
CONFIG5H	0x1100 0x0000	0xC0	
CONFIG6L	0x1111 0x1111	0xFF	0xE0FF
CONFIG6H	0x1110 0x0000	0xE0	
CONFIG7L	0x1111 0x1111	0xFF	0x40FF
CONFIG7H	0x0100 0x0000	0x40	
DEVID1	0x0000 0x0000	0x00	0x0600
DEVID2	0x0000 0x0110	0x06	

As principais configurações da Pilha TCP/IP são:

- Oscilador: HS
- Watchdog Timer: Desabilitado
- Low Voltage Programming: Desabilitado

10. Apêndice A . Configurando o Windows XP com IP fixo.

Nosso objetivo neste apêndice é configurar o computador com endereço IP 10.0.0.100 e máscara de rede 255.255.255.0. Para isso, siga os seguintes passos:

1. No menu Início, selecione o Painel de Controle.
2. Dê um duplo clique em Conexões de Rede.
3. Clique com o botão direito do mouse em Local Área Connection e selecione propriedades.
4. Selecione Internet Protocol (TCP/IP) e clique no botão Propriedades.
5. Configure o endereço IP com 10.0.0.100 e Máscara de Rede com 255.255.255.0 conforme Figura 6 .
6. Clique em OK.

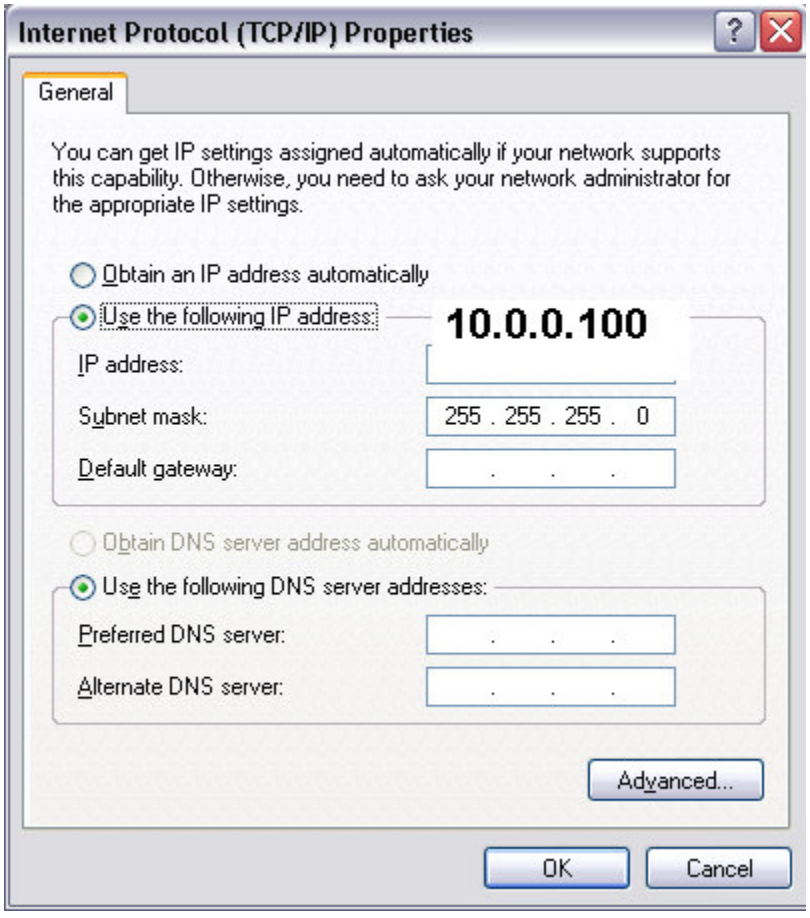

Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
10.0.0.100
IP address:
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

Figura 6 Configurando IP e Máscara de Rede

11.Apêndice B

A seguir listaremos os arquivos que fazem parte do projeto e suas respectivas macro definições que fazem parte do projeto MPNICPG.mcp. Estes arquivos podem ser vistos com o uso do menu View\Project. As macros podem ser acessadas clicando com o botão direito do mouse em cada arquivo e em seguida clicando-se em Build Options.

File	Macros
arp.c	**MPFS_USE_PGRM** **STACK_USE_HTTP_SERVER** **STACK_USE_DHCP** (só no uso do DHCP) USE_LCD (só quando do uso do LCD, pode travar o programa se não houver LCD)
ARPTsk.c	**IDEM**
C18Cfg.asm	**IDEM**
delay.c	**IDEM**
dhcp.c	**IDEM** (só deve estar presente no projeto quando do uso do DHCP)
Helpers.c	**IDEM**
http.c	**IDEM**
icmp.c	**IDEM**
ip.c	**IDEM**
Mac.c	**IDEM**
mpfs.c	**IDEM**
MPFSimg.c	**IDEM**
StackTsk.c	**IDEM**
tcp.c	**IDEM**
Tick.c	**IDEM**
udp.c	**IDEM** (só deve estar presente no projeto quando do uso do DHCP) **CELULAR_SMS** para enviar e receber mensagens via celular
websrvr.c	**IDEM**
xlcd.c	**IDEM**

Para IP fixo a definição STACK_USE_DHCP deve ser retirada. Quando do uso do DHCP a macro STACK_USE_DHCP deve ser adicionada a "Macro Definitions".

12.Apêndice C. O Servidor HTTP Microchip

O Servidor HTTP da Microchip utilizado em nossa aplicação é implementado como uma tarefa cooperativa que co-existe com a pilha Microchip TCP/IP e a aplicação principal do usuário. O Servidor é implementado pelo arquivo "HTTP.C" , com a aplicação do usuário implementando duas chamadas de funções. O arquivo de demonstração "Websrvr.c" deverá ser usado como modelo de aplicação para criação das interfaces apropriadas.

O Servidor HTTP da Microchip não implementa todas as funcionalidades dos documentos RFC, mas possui características mínimas para implementação em sistemas embarcados. O usuário pode facilmente adicionar novas funcionalidades como desejar.

O Servidor HTTP incorpora as seguintes características:

- Suporta múltiplas conexões HTTP
- Contém *Simple File System* (MPFS)
- Suporta páginas Web localizadas na memória Flash do microcontrolador
- Páginas Web compatíveis com o formato MPFS
- Suporta o método HTTP "GET" (outros métodos podem ser facilmente adicionados)
- Suporta *Common Gateway Interface* (CGI) para invocar funções pré-definidas pelos navegadores
- Suporta geração de conteúdo dinâmico de páginas Web

Qualquer página Web a ser armazenada na PME-10 dever ser convertida para o formato MPFS.

O Servidor HTTP usa o arquivo "index.htm" como a página Web inicial a ser chamada pelo navegador Web, através do comando http://X.X.X.X (aonde X.X.X.X é o endereço IP da placa PME-10, seja estático ou dinâmico). Então todas as aplicações devem incluir um arquivo "index.htm". Se necessário, o nome desta página inicial pode ser trocada modificando a definição do compilador HTTP_DEFAULT_FILE_STRING no arquivo "http.c". É muito importante que nenhuma página Web contenha algum dos seguintes caracteres:

- ' ou "
- < ou >
- #
- %

- {, }, [ou]
- |
- \
- ~
- ^

Se uma das páginas Web contiver um destes caracteres, a correspondente página Web tornar-se-á inacessível. Nenhum aviso de atenção será dado.

O Servidor HTTP mantém uma lista de tipos de arquivos que ele suporta. Esta informação é usada para avisar um navegador como interpretar um arquivo em particular, baseado na extensão do arquivo. Por padrão, o Servidor HTTP Microchip suporta arquivos ".txt", ". htm", ".gif", ".cgi", ".jpg", ".cla" e ".wav". Se uma aplicação usa tipos de arquivos que não estão incluídos nesta lista, o usuário pode modificar a tabela "httpFiles" através da correspondente enumeração "httpContents" no arquivo "http.c".

O servidor HTTP pode alterar páginas dinamicamente e substituir informações em tempo real, como status de entradas e saídas do microcontrolador. Para incorporar estas informações em tempo real, o correspondente arquivo CGI (*.cgi) deve conter uma string '%xx', aonde o caracter '%' serve como código de controle e 'xx' representa o identificador da variável com dois dígitos.

```
<html>
<meta http-equiv="refresh" content="3">
<body>

<table>
    <tr>
        <td><b>Resultados</b></td>
    </tr>
    <tr>
        <td>Pot1:</td>
        <td>%02</td>
    </tr>
    <tr>
        <td>Pot2:</td>
        <td>%03</td>
    </tr>
</table>
```

O identificador das variáveis deve estar entre 00-99. Quando o Servidor HTTP encontra %xx, ele remove o caracter % e chama a função HTTPGetVar. Se uma página requer o caracter '%' ele deve ser precedido

por outro caracter '%'. Por exemplo, para mostrar "23%" em uma página, coloque "23%%".

A seguir explicaremos os parâmetros da função HTTPGetVar e exemplificaremos um pequeno código contido nesta função.

HTTPGetVar(Byte var, Word ref, Byte *val)

var: ENTRADA . Identificador da variável cujo valor é para ser retornado.

ref: ENTRADA. Usado para uma única transferência ou transferência de vetores.

val: SAÏDA. Byte a ser transferido.

Valor de retorno. Se o valor retornado for diferente de HTTP_END_OF_VAR, o servidor HTTP chamará a função outra vez. Se o valor retornado for HTTP_END_OF_VAR o servidor HTTP não mais chamará a função e assume que a variável foi totalmente transferida.

No Exemplo 1 (Figura 7) a seguir, o parâmetro **val** representa o estado da porta RB5. Note também que o retorno HTTP_END_OF_VAR indica que a variável é de um byte somente.

```
WORD HTTPGetVar(BYTE var, WORD ref, BYTE *val)
{
        // identificação da variável
        // é referente a RB5
        if ( var == 4 )
        {
                // retornaremos '1 ' se RB5 estiver em HIGH, ou '0'se LOW
                if ( PORTBbits.RB5 )
                        *val = '1';
                else
                        *val = 0;
                // Reporta ao HTTP que este foi o último byte do valor da variável
                return HTTP_END_OF_VAR;
        }
        // Checa outras variáveis
```

Figura 7 Exemplo 1

No Exemplo 2 (Figura 8) a seguir, mostraremos o número serial do Servidor Web.

```
WORD HTTPGetVar(BYTE var, WORD ref, BYTE *val)
{
        // identificação da variável
        // é referente a RB5
        // Se sim, segue o mesmo do Exemplo 1
...
...
        // é referente a variável do número serial
        if ( var == 5)
        {
                // o número serial acaba com um caracter NULL
                // sendo a primeira chamada
                if (ref == HTTP_START_OF_VAR)
                {
                        // Esta é a primeira chamada. Inicializa índice para n. série
                        ref = (BYTE)0;
                }
                // Agora acessamos byte no índice corrente e salvamos no buffer
                *val = SerailNumberStr[(BYTE)ref];
                // É final da string
                if (*val=='\0')
                {
                        //Sim, nós estamos transferindo a string.
                        //Retornamos com HTTP_END_OF_VAR para notificar o
                        //servidor HTTP que nós estamos finalizando a transferência
                        //do valor
                        return HTTP_END_OF_VAR;
                }
                // Ou caso contrário, incrementa o índice do vetor e retorna para o
                // servidor HTTP
                (BYTE)ref++;
                // desde que não é o fim da string retorna  ref
                return ref;
        else
        // checa por outras variáveis
```

Figura 8 Exemplo 2

A página "status.cgi" sendo servida pelo servidor HTTP contém a seguinte linha HTML:

....
<td>Serial Number=%05</td>
....

O servidor HTTP ao processar este arquivo e encontrando a string '%05' faz uma chamada HTTPGetVar(4, HTTP_START_OF_VAR, &value). A aplicação principal implementa HTTPGetVar como segue no Exemplo 2.

Os Exemplos 1 e 2 mostraram em um navegador Web variáveis que se encontram na memória do microcontrolador. O Exemplo 3 mostra como alterar valores em um navegador Web no cliente e transferi-los para a memória do microcontrolador.

No navegador Web a seguinte página (Figura 9 e Figura 10) permite a alteração da variável "Nível de Potência", "Limite de Potência Inferior" e "Limite de Potência Superior".

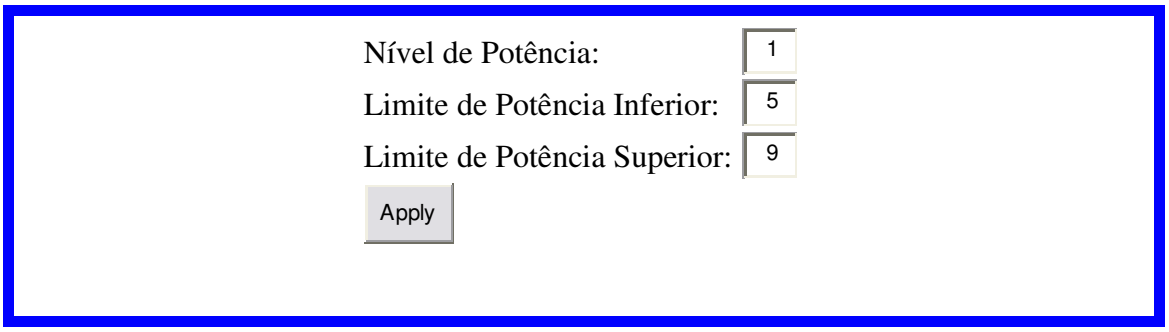


Figura 9 Página Web permitindo alteração de variáveis

```
<html>

<body><center>
<FORM METHOD=Get action=Power.cgi>
<table>
<tr><td>Nível de Potência:</td>
<td><input type=text size=2 maxlength=1 name=P></td></tr>
 <tr><td>Limite de Potência Inferior:</td>
<td><input type=text size=2 maxlength=1 name=L ></td></tr>
<tr><td>Limite de Potência Superior:</td>
<td><input type=text size=2 maxlength=1 name=H ></td></tr>
<tr><td><input type=submit name=B value=Apply></td></tr>
</table>
</body>
</html>
```

Figura 10 Código da página Web que permite alteração de variáveis (Commands.cgi)

Esta página mostra uma tabela com o nome das variáveis na primeira coluna e caixa de textos para entrada de valores na segunda coluna. A primeira linha, primeira coluna contém o nome da variável "Nível de Potência"; a segunda coluna é uma caixa de texto para mostrar e modificar o nível de potência. A última linha contém um botão "Apply" . Com esta página o usuário tem a habilidade de modificar o nível de potência na caixa de texto e enviar este valor ao microcontrolador na PME-10.

Assuma que o usuário entra com os valores '5','1'e '9'respectivamente nas caixas de texto do nível de potência, limite inferior de potência e limite superior de potência . Em seguida clica no botão "Apply". O navegador cria uma requisição HTTP com uma string "Power.cgi?P=5&L=1&H=9" e envia ao servidor HTTP. O servidor chama a função HTTPExecCmd com os seguintes parâmetros:

argv[0]="Power.cgi",argv[1]="P",argv[2]="5",argv[3]="L',argv[4]="1",argv[5]="H", argv[6]="9" argc=7

A aplicação principal deverá implementar a função HTTPExecCmd como segue (Figura 11):

```
void HTTPExecCmd(BYTE *argv, BYTE argc)
{
        BYTE i;
        // Varre todos os parâmetros
        for (i=1; i < argc;i++)
        {
                // Identifica parâmetros
                if ( argv[i][0] == 'P')          // É nível de potência?
                {
                        PowerLevel = atoi(argv[++i]);
                }
                else if ( argv[i][0] == 'L' )    // É limite inferior de potência?
                        LowPowerSetting = atoi (argv[++i]);
                else if (argv[i][0] == 'H')      // É limite superior de potência?
                        HighPowerSeting = atoi (argv[++i]);
        }
        // se outra página é para ser mostrada como resultado deste comando
        // copie o nome em maiúsculo para argv[0]
        // strcpy(argv[0], "RESULTS.CGI");
}
```

Figura 11 Função HTTPExecCmd alterando vaolores de variáveis

Neste exemplo, o número total de argumentos excede o padrão de 5. Então modifique o valor de MAX_HTTP_ARGS (localizado em "http.c") para pelo menos 7.

13. Apêndice D. Enviando e recebendo mensagens SMS através do Celular Siemens C55 (Modo PDU).

Nosso exemplo consiste em enviar os caracteres ".b01" para o telefone celular de número 91897299. ".b01" significa que o microcontrolador deverá colocar em nível lógico alto a saída 1 da porta B.

Os seguintes procedimentos devem ser seguidos:

- Apague todas as mensagens que se encontram em seu celular, sejam enviadas ou sejam recebidas.
- Envie através de seu celular uma mensagem contendo somente o caracter "A" para o número do celular desejado (em nosso caso 91897299).
- Conecte o cabo serial entre um computador PC e o telefone celullar Siemens C55. Faça uma conexão com o celular usando o HyperTerminal com configuração da conexão 9600, 8, N, 1.
- Execute o comando **AT** e verifique como resposta **OK**.
- Execute o comando **AT + CMGL=4**.
- Verifique que a resposta deve ser algo parecido com :

+CMGL : 3,2,,13
07915591191035 00B10008811998279900 00AD0141

Significados:
O algarismo **3** significa que foi uma mensagem enviada.
O algarismo **2** significa a posição de memória em que se encontrava a mensagem.
O algarismo 13 significa o comprimento da mensagem sem considerar o número do centro de serviço de mensagens (**559119103500**), o comprimento do número do centro de serviço de mensagens (**07**) e o tipo de endereçamento (**19**).

Os bytes da segunda linha significam:

1	07	Comprimento do número do centro de serviços incluindo tipo de endereço
2	91	tipo de endereço
3	55	Número do centro de serviço. O número do centro de serviço é 551991015300. Osbserve que a codificação é 559119103500. Você pode verificar o número do centro de serviços através do comando AT + CSCA?
4	91	
5	19	
6	10	
7	35	
8	00	
9	B1	Mensagem vai do telefone para a rede
10	00	TP-Message-Reference
11	08	Comprimento do número do telefone a ser enviada a mensagem
12	81	International Adress
13	19	Número do telefone 91897299. Verifique que a ordem da codificação é 19982799.
14	98	
15	27	
16	99	
17	00	TP-Protocol-Identifier
18	00	TP-Data-Coding-Scheme
19	AD	TP-Validaty-Period
20	01	Comprimento da mensagem cacarcteres ASCII em hexadecimal
21	41	caracter "A "

- Para enviar os caracteres ".b01" para o telefone 91897299 devemos executar os seguintes comandos no Hyperterminal:

  AT + CMGC=?
  Mensagem recebida **OK**
  AT + CMGF = 0
  Mensagem recebida **OK**
  AT+CMGS=16 (Obs: 16= 12 + tamanho da mensagem em octetos em decimal)
  Mensagem recebida **>**

>079155911910350OB1000881199827990000AD042E312C06<ctrl-Z>
Mensagem recebida **+CMGS: 62**
OK

Como transformar os caracteres ".b01" do formato ASCII para o formato de octetos.

. = 0x2E = (0x **0010 1110**)
b = 0x62 = (0x **0110 0010**)
0 = 0x30 = (0x **0011 0000**)
1 = 0X31 = (0X **0011 0001**)

Devemos somente levar em consideração os 7 bits menos significativos de cada caracter, de modo que temos:

. = 0x2E = (0x **010 1110**)
b = 0x62 = (0x **110 0010**)
0 = 0x30 = (0x **011 0000**)
1 = 0X31 = (0X **011 0001**)

O primeiro octeto a ser transmitido é formado pelo bit menos significativo do caracter "b" e os 7 bits do caracter "." = 0x **0 0101110** = **0x2E** .

O segundo octeto a ser transmitido é formado pelo dois bits menos significativo do caracter "0" e os 6 bits do caracter "b" que não foram usados no primeiro octeto = 0x **00 110 001** = **0x31**.

O terceiro octeto a ser transmitido é formado pelo três bits menos significativo do caracter "1" e os 5 bits do caracter "0" que não foram usados no segundo octeto = 0x **001 011 00** = **0x2C**.

O quarto octeto a ser transmitido é formado pelos quatro bits do caracter "1" que não foram usados no terceiro octeto complementados com zeros a esquerda = 0x 0000 **011 0** **= 0x06**.

Em relação aos dados reais da mensagem temos que transmitir: **0x2E 0x31 0x2C 0x06.**

Obs: Se executarmos o comando AT+CMGF? e recebermos como resposta o parâmetro "1" também, significa que o seu celular pode enviar uma mensagem no formato texto.

Comando AT	Descrição
AT+CMGF=1<enter>	SMS em modo texto.
AT+CMGS="+4441793181022"<enter> <texto><ctrl–Z>	A mensagem é enviada para o número +4441793181022. Substitua o núemro +444... com seu próprio número.
	Mensagem enviada com sucesso.

No programa , arquivo , verifique as seguintes linhas de programa:

Exemplo 2

Mensagem a ser enviada: **Tanque setor norte\r\n**
alarme ativo\r\n
necessita manutenção

Número de caracteres: **54**
Número de bits = 54 * 7 = 378 bits
Número de octetos na mensagem = 378/8= 47 octetos + 2 bits ➔ 48 octetos
(complementa os últimos 2 bits com zeros)

Em relação aos dados reais da mensagem temos que enviar:

F4 B0 3B 5E 2F 83 E6 65 FA 5B 0E 72 BF E5 F4 32 48 11 66 87 E5 ED 32 28 4C 4F
DB DF 20 85 BB 3C 2E CF E7 69 7A 18 D4 0E BB EB F4 B2 7B 1C 7E 03

Obs: verifique a existência de 48 octetos

Os comandos a serem enviados ao modem devem ser:

AT+CMGC=?
AT+CMGF=0

AT+CMGS=60 (12 + número de octetos da mensagem em decimal) = (12 + 48)=60

Oct.		Descrição
0	07	Comprimento do número do centro de serviços incluindo tipo de endereço
1	91	tipo de endereço
2	55	Número do centro de serviço. O número do centro de serviço é 551991015300. Osbserve que a codificação é 559119103500. Você pode verificar o número do centro de serviços através do comando AT + CSCA?
3	91	
4	19	
5	10	
6	35	
7	00	
1	B1	Mensagem vai do telefone para a rede
2	00	TP-Message-Reference
3	08	Comprimento do número do telefone a ser enviada a mensagem
4	81	International Adress
5	19	Número do telefone 91897299. Verifique que a ordem da codificação é 19982799.
6	98	
7	27	
8	99	
9	00	TP-Protocol-Identifier
10	00	TP-Data-Coding-Scheme
11	AD	TP-Validaty-Period
12	36	Comprimento da mensagem em ASCII em HEXA
13-57		Mensagem **74 2f f2 bf 43 e5 4c 0d 2e ba bb fb b1 83 5b e5 21 ed 4f 85 cf 18 eb 3e 3b e6 0e f4 66 32 db bb e7 d4 f4 7e 5e 65 72 72 8b 48 df 3c 69 15 b2 03**

AT + CMGC=?
Mensagem recebida **OK**
AT + CMGF = 0
Mensagem recebida **OK**
AT+CMGS=60
Mensagem recebida **>**

>0791559119103500B10008811998279900000AD3674 2f f2 bf 43 e5 4c 0d 2e ba bb fb b1 83 5b e5 21 ed 4f 85 cf 18 eb 3e 3b e6 0e f4 66 32 db bb e7 d4 f4 7e 5e 65 72 72 8b 48 df 3c 69 15 b2 03<ctrl-Z>

Mensagem recebida **+CMGS: 62**

OK

A 2EI implementou uma função denominada **ler_mensagem_celular** que é capaz de setar portas do Microcontrolador PIC18F8720. A mensagem a ser enviada deve ser conforme exempo a seguir:

Mensagem SMS : **.a21**

- "." - caracter de controle
- "a" – porta do microcontrolador a ser programada
- "2" – bit da porta a ser programada
- "1" – estado da porta

Para enviar uma mensagem SMS deve-se usar a função **envia_SMS**. As seguintes constantes de memória FLASH devem ser alteradas:

ROM char MENSAGEM1[]
ROM char AT_TAMANHO[]
ROM char AT_PDU[]

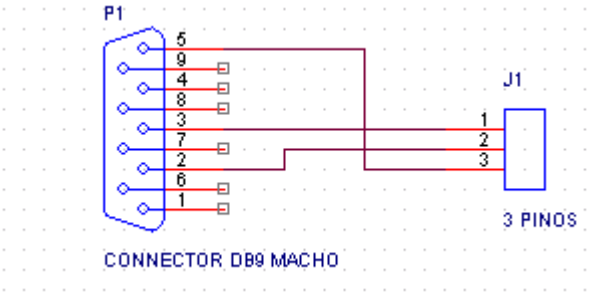
Nossos testes foram feitos com um celular Siemens modelo C65 enviando a mensagem SMS e um celular Siemens modelo C55 recebendo a mensagem na PME-10, operadora CLARO.

14.Apêndice E. DISPLAY DE LCD

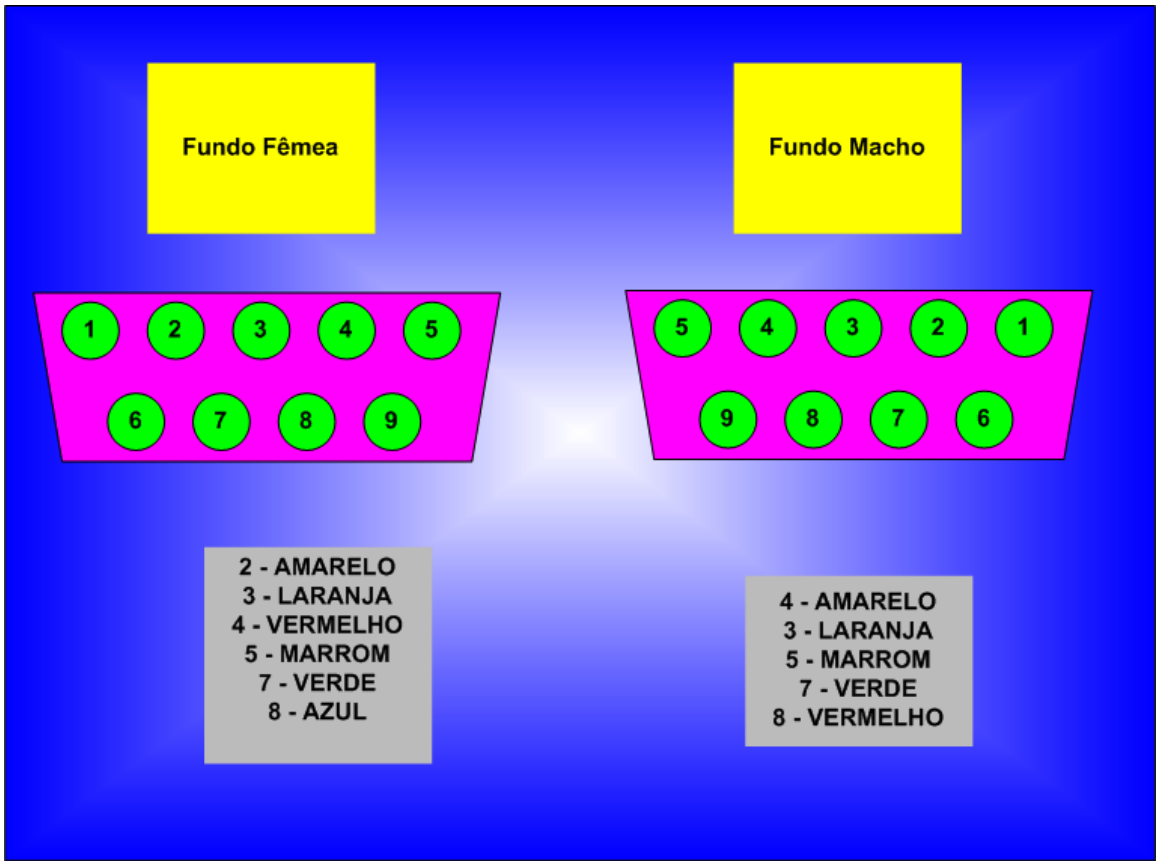
Ligação do Display de LCD POWERTIP modelo PC 1602-F

LCD POWERTIP modelo PC 1602-F				
Pino	Símbolo	Função	Porta	Conector
1	Vss	Power supply (GND)	GND	J4-5
2	Vdd	Power Supply (+ 5V)	+5VDC	J4-3
3	Vo	Contrast adjust	GND	J4-6
4	RS	Register select signal	RJ4	J4-4
5	R/W	Data read / write	RJ5	J4-1
6	E	Enable signal	RJ6	J4-2
7	DB0	Data bus line		
8	DB1	Data bus line		
9	DB2	Data bus line		
10	DB3	Data bus line		
11	DB4	Data bus line	RJ0	J4-8
12	DB5	Data bus line	RJ1	J4-7
13	DB6	Data bus line	RJ2	J4-10
14	DB7	Data bus line	RJ3	J4-9
15	A	Power supply for LED B/L(+)	+5 VDC	
16	K	Power supply for LED B/L (-)	GND	

15.Apêndice F. CABO SERIAL CELULAR



16.Apêndice G. CABO SERIAL PROGRAMAÇÃO



17.Apêndice H. Diferenças para Programar o PIC18F8720 e o PIC18F8621

Diferenças no MPLAB IDE:

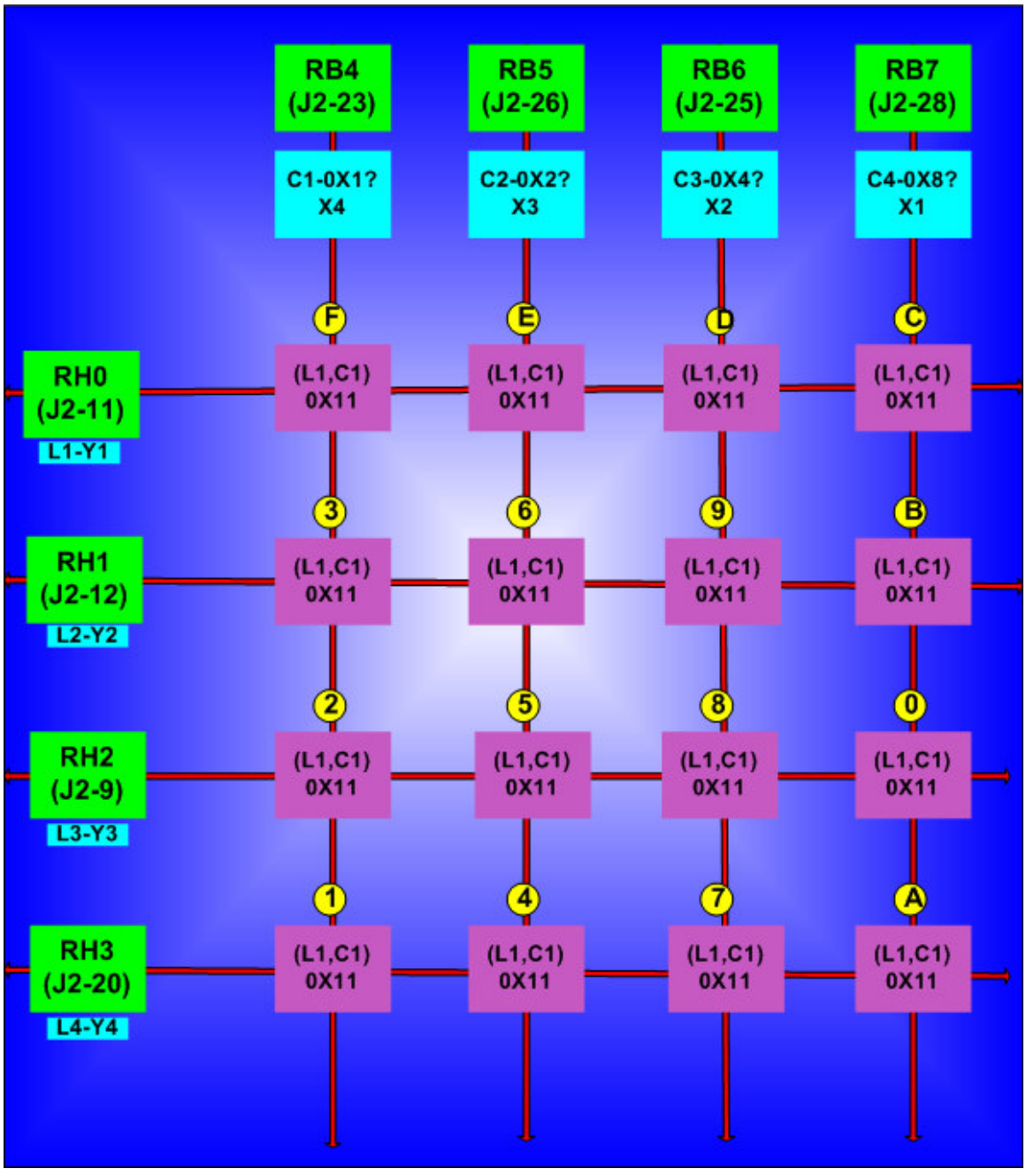
1. Header Files no menu View/Project: p18f8621.h
2. Linker Scripts no menu View/Project: 18f8621.lkr

Diferenças no IC-PROG:

1. O dispositivo selecionado passa a ser o PIC18F8620.

Quando usando o ICD da Mosaico no modo "Debugger" o arquivo de link deve ter a opção 18f8720i.lkr ou 18f8621i.lkr .

18.Apêndice I. Teclado



19.Apêndice J. Cabo de Rede

Chanfro para baixo	
Verde Claro	Branco
Verde	Laranja
Branco	Verde Claro
Azul Escuro	Azul Escuro
Azul Claro	Azul Claro
Laranja	Verde Escuro
Marrom Claro	Marrom Claro
Marrom	Marrom

20.Apêndice K. Alterações na versão deste Manual

V1.3	Macro CELULAR_SMS no arquivo WEBSRVR.C
V1.3	Alterações do nome nas etiquetas do cabo de programação. Para melhor entendimento passam a ser: .INT para interface de programação e PME10 para a placa PME-10.
V1.3	Não inverter o cabo de programação que vai da interface de programação para a placa PME-10.
V1.3	Indicação de como usar o IC-PROG no Windows XP.
V1.3	Endereço MAC próprio 2EI
V1.3	Diferenças para programar o PIC18F8621 e o PIC18F8720
V1.4	WinPIC para gravar PIC18F8722
V1.5	Teclado e cabo rede
V1.6	Comando winpic /device=xxxx
V1.7	Alteração do endereço IP 10.0.0.101. Durante a programação usando a interface IP10 jamais conecte qualquer outro terra a placa, por exemplo, cabo serial RS232.

21.Apêndice L. Servidor WAP

As indústrias que fabricam equipamentos sem fio desejavam padronizar um novo protocolo para comunicação entre equipamentos, assim surgiu a idéia do protocolo WAP. O objetivo principal deste padrão é mostrar conteúdo Internet em dispositivos sem fio, por exemplo, em telefones celulares. Podemos definir WAP como:

- WAP significa ***W**ireless **A**plicattion **P**rotocol*
- WAP é um protocolo de comunicação na camada aplicação
- WAP é usado para serviços e informações
- WAP é uma herança dos padrões Internet
- WAP é para dispositivos portáteis, como telefones celulares
- WAP é um protocolo projetado para micro browsers
- WAP possibilita a criação de aplicações web para dispositivos móveis
- WAP usa a linguagem de marcação WML (não HTML)
- WML é definida como uma aplicação XML 1.0

WML é uma linguagem de marcação que significa ***W**ireless **M**arkup **L**anguage*. É uma herança da linguagem de marcação HTML, mas WML é baseado em XML, então é bem mais consistente que HTML.

O processo para mostrar as informações em um dispositivo móvel pode ser visto na Figura 12.

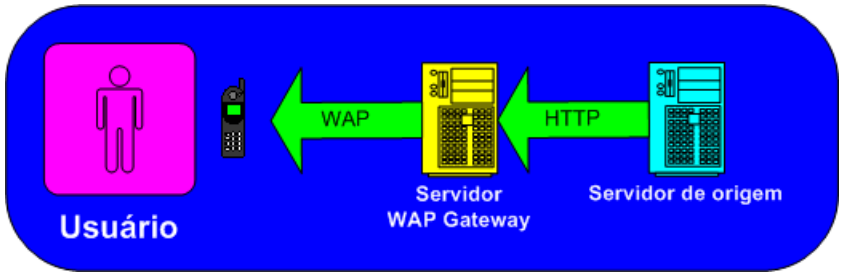


Figura 12: Transação WAP

O navegador WAP pode estar sendo executado em qualquer dispositivo móvel, desde um telefone celular até um PDA. O servidor de origem é aonde estão armazenadas as páginas WML e se comunica com o Gateway WAP através do protocolo HTTP. É usualmente necessário fazer modificações no servidor HTTP de origem para que ele possa armazenar conteúdo WAP(WML). O interessante é que o servidor de origem pode continuar usando tecnologia CGI para gerar qualquer conteúdo dinâmico. O trabalho do

gateway WAP é converter protocolos de comunicação do navegador WAP (WSP, WTP) para protocolos do servidor de origem, que podem ser HTTP e TCP/IP.

A seguir vamos descrever os passos para criação das páginas WML que mostrarão os valores das variáveis do microcontrolador.

WML é usada para criar páginas que podem ser mostradas num navegador WAP. Páginas em WML são chamadas DECKS. Decks são construídos com um conjunto de CARDS.

Nosso objetivo é mostrar as seguintes páginas WML no celular:

Figura 13: Página WML de recpção

Figura 14: Página WML com as variáveis

O navegador WAP pode ser baixado do site WinWap Technologies www.winwap.com .

A primeira página, arquivo index.wml, Figura 13, tem o código listado abaixo:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC
   "-//WAPFORUM//DTD WML 1.1//EN"
   "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
   <card title="2EI">
      <p>Ola, Usuario!</p>
                  <p>
                        <anchor>next page<go href="wmls1.cgi"/></anchor>
                  </p>
         </card>
</wml>
```

A segunda página, arquivo wmls1.CGI, Figura 14, tem o código listado abaixo:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC
   "-//WAPFORUM//DTD WML 1.1//EN"
   "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
 <card title="2EI - Eletronica Embarcada para Internet">
      <p>Pot1 = %02</p>
      <p>Pot2 = %03</p>
      <p>Chave = %04</p>
      <p>Led A2 = %01</p>
      <p>Serial = %06</p>
      <p>Teclado = %07</p>
      <p>Temperatura = %08</p>
 </card>
</wml>
```

As páginas WML são armazenadas na mesma pasta em que se encontram as páginas HTML e interfaces CGI. A chamada da página pelo navegador WAP deve ser feita da forma **"http://10.0.0.103:8081/index.wml".** No celular simplesmente digite "10.0.0.103:8081/index.wml". Se a placa PME-10 estiver ligado em um roteador

verifique o seu IP público e re-direcione a porta chamada para a porta 8081 ou que você programou.

Os arquivos .CGI que são chamados por páginas WML devem ser inicializados com as letras WML, a fim de que o servidor HTTP saiba que a chamada veio proveniente de uma página WML.

Em outra versão do manual colocaremos como enviar conteúdo dinâmico do celular para a PME-10.